

Introduction to Linux

A selection of articles from Both.org

Copyright © 2026

Introduction to Linux. Both.org contributors. Jim Hall,
Lily Yang (eds). Published under a Creative Commons Attribution ShareAlike license.

Thanks to the Both.org community!

Contents

Introduction.....	5
3 Surprising Reasons I Like the Cinnamon Desktop.....	7
5 Reasons for Windows Users to Move to Linux.....	13
5 Reasons to Love Init.....	25
5 Unix Books You Must Read.....	31
10 Reasons to Use Linux.....	43
A Brief History of the Linux Desktop.....	53
A Fresh Look at Manjaro KDE Plasma.....	69
Best Linux Distros for Windows Users: From Mint to Pop! _OS.....	73
Breaking Free from Windows 10: KDE Plasma Is Your Upgrade to Freedom.....	81
Choose your new computer's operating system.....	85
Cinnamon: A Tale of Forks and Familiarity in the Linux Desktop Landscape.....	89
Getting Help for Linux.....	97
Linux and Security.....	103
Linux Desktops and Laptops.....	107

My First Linux System.....	113
System76 and the New COSMIC Desktop.....	131
What is Linux?.....	139
Why I Switched to Linux.....	147

Introduction

Both.org is a group of writers dedicated to exploring the world of open-source software. As a way to celebrate our community, we collected a total of 18 different articles from the site that shows a passion for Linux.

Linux is a well-known open source platform, yet if you're new to it, you might not have even heard of it. Or maybe you have heard of Linux but want to learn more about it. Well then, this is a book you might want to read.

The book is an introduction to Linux for users and readers who are just getting started with Linux software. In this book, you'll find many articles that share great suggestions on how to start learning Linux. You'll find tips you can use, advice from experts, and introductions to different desktops—such as Cinnamon, KDE, Xfce, and GNOME—that you can use to explore further, based on your own personal preferences and needs.

By the end of this book, you'll know the basis of Linux and even what desktops you are most interested in to start your Linux journey.

3 Surprising Reasons I Like the Cinnamon Desktop

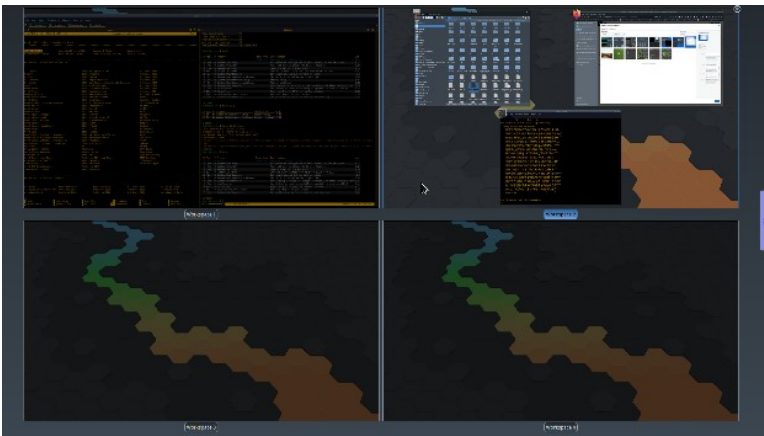
David Both
July 25, 2025

Since the last time I've written articles about Cinnamon, I've found even more reasons to like and use this surprisingly robust desktop. Cinnamon is a fork of the GNOME desktop before it morphed into the "non-desktop metaphor" interface it has today. The current release is Cinnamon 6.4.9.

Its interface is much like that of the early GNOME 2 desktop. It's simple and uncluttered and makes a good desktop for users migrating from Windows. For that reason, Cinnamon is also the default desktop for the Linux Mint distribution. It's also available for many other distributions, including my own favorite, Fedora. Today's Cinnamon brings a simple desktop experience into the modern era with some surprising features that are typically found in massively configurable environments like KDE. Such desktops also bring significant resource consumption along with their power.

1. Hot Corners

One of the features I like most about Cinnamon is its hot corners. When configured, moving the mouse to the appropriate corner of the display can show all the open windows on the current workspace, and you can select a window with the click of your mouse.



The Cinnamon desktop is using the hot corners feature to display all four workspaces.

In another corner it can display all workspaces with the windows open in each. A mouse click can take you to any desktop or any desired window on any workspace. This is incredibly useful for finding open windows—even when minimized—that might be hiding in plain sight.

A third corner can be configured to minimize all windows on all workspaces, and a fourth can be used to launch a preset command.

This feature is much like that in KDE Plasma 6 and is one of the reasons I now prefer Cinnamon over other desktops, including Plasma and Xfce.

2. Lock Keys Indicator

One consistent issue I have with my laptops and some small keyboards I use on other systems is that of knowing the status of the Num Lock and Caps Lock keys. Several of my small keyboards and that of one of my laptops have no LED indicators at all for the lock keys. My other laptop does have an LED indicator, but it's tiny and can't be distinguished from other tiny LED indicators.

But there's an applet for that. Open "System Settings," select "Applets," and then the "Download" button. Install the "Lock keys indicator with notification" applet, and configure it to display Num Lock and Caps Lock status and notifications. When activated, this applet displays a small set of icons in the system tray. The box containing an "A" represents Caps Lock, and the box containing "1" is for Num Lock. Each box is highlighted when that key is active and a notification is displayed on the workspace whenever the status of one of the keys changes.

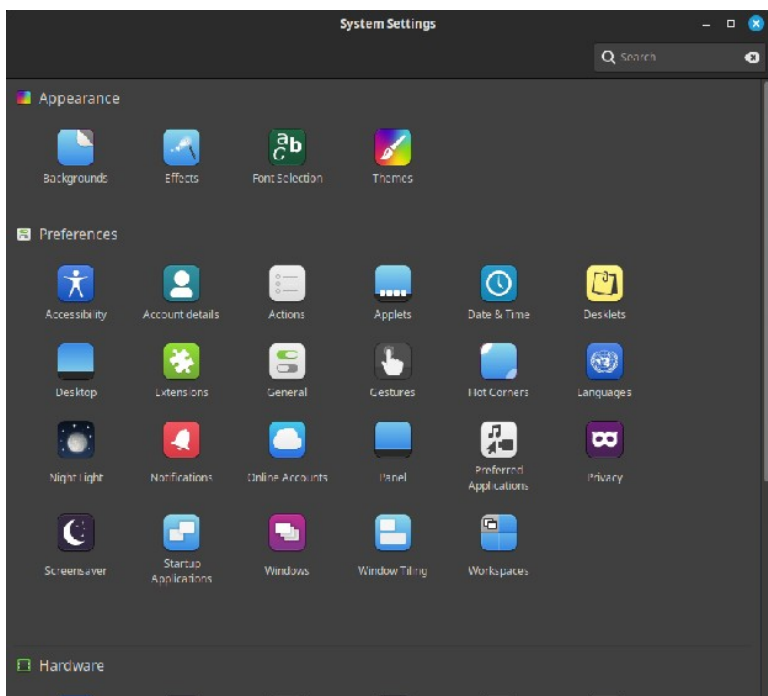
3. The Cube

This feature is pure fun, as it simulates a cube that rotates as you switch between the workspaces. It's a nice touch and offers an enticing look at the powerful plugin and extension capabilities of the Cinnamon desktop.

Add-ons

These fun and useful features are only three of the many available for Cinnamon that are only possible because of its designed-in extensibility.

Like most desktop environments, the look and feel of Cinnamon can be modified by any of the many different themes that are available in the original installation or via downloads. Themes can allow users to personalize the Cinnamon interface components such as the launch menu, the panel, its built-in calendar, and various dialogs.



The Cinnamon System Settings dialog allows you to download and install themes, applets, desklets, actions, and extensions.

Applets are icons or text dialogs that appear on the system panel. The main Cinnamon menu is one of these applets. Others include the calendar, an accessibility menu, a Network-Manager applet that allows users to view and manage network adapters and connections, and many more fun and useful tools.

Desklets are little tools that can be placed on the desktop, such as a digital clock and a mini-photo frame, which are two of the three provided by default. There are 64 of these desklets avail-

able at the time of this writing and they include a weather app and one that shows your Google calendar.

Actions are tasks like converting files from one format to another, taking screenshots, backing up files, installing fonts, and many more.

Extensions can modify some of the basic functions of Cinnamon. This is where you can download the desktop cube animation, wobbly windows, and other fun and useful effects.

You can go to Cinnamon Spices [<https://cinnamon-spices.linuxmint.com/>], the repository that contains many of these add-ons that you can download and try. That's a good place to get an overview of all the add-ons that are available, but they can all be downloaded and installed directly from the Cinnamon System Settings menu icons with the corresponding names.

Be prepared to invest a significant amount of time exploring these add-ons, as I've found myself spending hours there.

5 Reasons for Windows Users to Move to Linux

David Both
April 14, 2025

As the October deadline for dumping Windows 10 is fast approaching, many users are considering their options. This is for many reasons but is generally rooted in the ever-increasing enshittification of Windows.

There are also many articles out there that list all the reasons why Windows is so terrible, and we've published a few of those on Both.org. But I want to switch the viewpoint from Windows to Linux. I'll still be hard on Windows, but I'll be exploring the strengths of Linux.

1: Linux Respects Your Privacy

Linux doesn't require personal information or any type of account just to install. You can just download it and install it on as many computers as you want. There are no product keys to enter and no personal information of any kind. None of your data will be automatically transmitted to a cloud unless you want it to be and set it up yourself.

These items of personal information are *required* by Microsoft in order to be *allowed* to install Windows.

1. Name and surname
2. Birthdate
3. Country/region
4. Region

No matter that you've already purchased the product and have a computer capable of running it, this information is allegedly needed "to enhance security and user experience of Windows 11." Of course they "allow" users to enter additional information. The complete list of information they're trying to collect consists of nine items.

1. Name and surname
2. Birthdate
3. Sex
4. Country/region

5. Region
6. Postal code
7. Time zone
8. Phone
9. Address

That looks to me like enough to steal your identity or to at least convince you that they're calling from your bank or credit card company as they go phishing for your account data.

Linux has no Orwellian spy software like Recall [<https://arstechnica.com/security/2025/04/microsoft-is-putting-privacy-endangering-recall-back-into-windows-11/>] that snapshots your screen—*your screen*—with *your data* displayed every few minutes. This data includes everything that you see on your screen. Personal and confidential data such as financial and medical pictures—including the ones you don't want anyone else to see—legal documents, work products for your job, and more are being captured and sent to the cloud as an image, just as if someone had taken a picture of your screen and posted it on social media. In this case, the Microsoft cloud. This feature is allegedly so that you can—ummm—recall it to see what you were doing on your computer in the past and recover the data from a point in time. That also assumes that their AI can correctly scrape the information off the snapshot of your screen. And we all know how reliable and secure AI is.

And, of course, your privacy with Recall is totally dependent upon Microsoft being able to prevent bad actors from cracking into the Microsoft cloud.

Even with opt-in being required for Recall to be activated, this is a scary proposition.

2: Linux is Highly Secure

I've been using Linux since 1996, and—with one exception—I've never had a security breach, malware infection, or ransomware on any of my Linux hosts.

That exception was on one website on my server that I was not diligent about ensuring that WordPress updates were properly installed on. However, I discovered it quickly and recovery took about an hour to complete. None of the other websites were affected. The malware was introduced to that website using a known vulnerability that had a patch available but that had not been installed by the WordPress automated update process.

That's one single infection in 49 years and it was *my own fault* that I didn't check that those updates were being properly processed.

Linux is locked down right from the initial installation. It has a firewall that is set to block all incoming attempts to connect to your computer. There aren't any processes running that would

provide points of entry, even if the firewall were to fail and be breached. Updates, including security updates, are provided as a stream when they become available rather than once a month on a Tuesday. You can choose when to install updates and continue to use your computer while the upgrade is taking place. You can also choose when to perform a reboot if one is even required, which is not always the case.

Of course, I do recommend that updates be installed on a regular basis and that you reboot your Linux computers when necessary. Some distros, Fedora for one, include an easy-to-run tool that can tell you if a reboot is required.

Linux has many open-source and free (as in beer) options for preventing and detecting malware on your computer. You can install as many of those as you need to feel comfortable. This includes tools like Tripwire [https://en.wikipedia.org/wiki/Open_Source_Tripwire], which informs you if a cracker¹ has managed to gain access to your system; chkrootkit [<https://www.chkrootkit.org/>] (Check RootKit), which can suss out rootkits that may have been installed by a cracker; and ClamAV [<https://www.clamav.net/>] antivirus software, which can detect viruses that have infected your system as well as scan incoming email for viruses. Many of us Linux users depend on these tools to verify the continuing integrity of our computers, despite the knowledge that they are extremely secure in the first place.

If you want the ultimate in protection, the NSA wrote an interesting bit of software and made it available to the open-source community as Security Enhanced Linux (SELinux) [https://en.wikipedia.org/wiki/Security-Enhanced_Linux]. It's not a distro; it's just a bit of software, a kernel module that can prevent changes from being made to the programs, and configuration files on your Linux computer. Even if someone does gain access, they can't do anything like install malware or modify your files. The NSA uses this tool to protect its own computers. SELinux has been part of the basic installation of many Linux distros for over 20 years.

Okay, so you're thinking that the NSA is a government spy agency and how do we know that they didn't install a back-door or other spyware with SELinux? Thousands of independent programmers around the world who specialize in security have examined the open-source code for SELinux and found no such code embedded in it.

3: Linux Offers Many Choices

The choices we have as Linux users are almost limitless. Of course the first choice is whether to use it or not. No one will force you to use Linux. You can choose to purchase a computer that has Linux on it or to install it on a computer that you already have.

The next choices you have are which desktop and distribution to use. The desktop is the graphical user interface (GUI) that allows you to interact with the computer. The distribution is a completely packaged group of programs and the Linux kernel that can be installed on your computer. There are many distributions like Linux Mint [<https://www.linuxmint.com/>], Ubuntu [<https://ubuntu.com/>], Arch [<https://archlinux.org/>], Slackware [<http://www.slackware.com/>], Fedora [<https://fedoraproject.org/>], and more. Each has a preferred desktop and set of application programs. However, most distributions provide alternate desktops for you to install and try.

No, I'm not going to tell you which desktop or distro to use. I stick with Fedora because I like it and I've used it for a long time. I use other distros like Linux Mint, Pop!_OS, and Ubuntu in VMs for testing and research. I do tend to switch between three or four different desktops on a regular basis. The three current ones I use are Xfce [<https://xfce.org/>], KDE Plasma [<https://kde.org/plasma-desktop/>], and Cinnamon [<https://projects.linuxmint.com/cinnamon/>]. Many more than these are available.

Applications are the reason we use computers. Whether it's to write books, articles, reports, or other documents; to create budget spreadsheets and keep track of business or personal finances; to create and edit audio and video programs; to plan your astronomical viewing session and implement it by aiming your telescope; to create and manage digital pictures and artwork; to browse the web; and so much more, Linux usually

has multiple applications for each task from which you can choose.

There are also hundreds of games available for Linux.

4: Linux is About Freedom

Freedom has an entirely different meaning when applied to free open-source software (FOSS) than it does in most other circumstances. In FOSS, free is the freedom to do what I want with software. It means that I have easy access to the source code and that I can make changes to the code and recompile it if I need or want to.

Freedom means that I can download a copy of Fedora Linux, or Firefox, or LibreOffice, and install it on as many computers as I want to. It means that I can share that downloaded code by providing copies to my friends or installing it on computers belonging to my customers, both the executables and the sources.

Freedom means that we do not need to worry about the license police showing up on our doorsteps and demanding huge sums of money to become compliant. This has happened at some companies that “over-installed” the number of licenses that they had available for an operating system or office suite. It means that I don’t have to type in a long, long “key” to unlock the software I have purchased or downloaded.

Freedom is the ability to choose when we install updates. We can also choose when to upgrade from one release level to the next, such as from Fedora 41 to Fedora 42.

Linux is free from pop-ups that are simply advertisements for more M\$ or “partner” products. It’s free from icons in the Applications menu that are for software that isn’t installed and which we must purchase to use. These are misleading and quite frustrating.

Freedom means that we have control over our operating system and the applications that we run on it, not some humongous organization whose only care is about profit.

“The value of any software lies in its usefulness, not in its price.”

— Linus Torvalds, creator of Linux

Our Software Rights

The rights to the freedoms that we have with open-source software should be part of the license we receive when we download open-source software. The definition for open-source software is found at the Open Source Initiative [<https://opensource.org/osd>] website. This definition describes the freedoms and responsibilities that are part of using open-source software.

The issue is that there are many licenses that claim to be open-source. Some are and some are not. In order to be true open-source software, the license must meet the requirements specified in this definition. The definition is not a license; it specifies

the terms to which any license must conform if the software to which it is attached is to be legally considered open-source. If any of the defined terms do not exist in a license, then the software to which it refers is not true open-source software.

I have not included that definition here despite its importance because you can go to the website previously cited. I strongly recommend that you at least go to the website and read the definition so that you will more fully understand what open-source really is and what rights you have.

I also like the description of Linux that originally appeared at [Opensource.com](https://opensource.com) and that we've reproduced here. [Opensource.com](https://opensource.com) has a long list of Linux and open-source resources [<https://opensource.com/resources>] that you should also check out.

5: Linux is Helping to Save the Planet

Linux is always an option for old computers, even the really old ones. I have one old Dell that will be 20 years old in August of this year. It has a rather minimal hardware configuration but is currently running a full install of Fedora Linux 24x7. And it's not just sitting there; I use it for real work, like research for my books, articles, and the World Community Grid [<https://www.worldcommunitygrid.org/>]. This old Dell is

also my primary test platform for my “How long can I keep an old computer running?” experiments.

The best option for the planet is to keep old computers running as long as possible. Replacing the defective components as they go bad can keep a computer running for years longer than the currently accepted lifespan and significantly reduces the amount of electronic waste that we dump in landfills or that needs recycling.

Linux can keep old computers running for as long as the hardware continues to work. Of course, most old computers can be fixed by replacing defective parts for far less than purchasing a new computer, but that’s a topic save for another time.

-
1. Free On-Line Dictionary Of Computing (FOLDOC) *cracker, cracking*. An individual who attempts to gain unauthorised access to a computer system. These individuals are often malicious and have many means at their disposal for breaking into a system.

5 Reasons to Love Init

Seth Kenlon
February 17, 2025

The init system of an operating system is the component that starts all the programs that run on your computer. Without init, you'd have to launch everything manually, including the interface you would use to do the launching! The init system is also the component that gracefully stops those processes when you send a shutdown signal to your computer.

In a previous article, I wrote about my favorite features of systemd. The truth is, I don't run systemd on all my computers, most notably, not on the desktop Slackware PC that I use daily and certainly not on my NetBSD Raspberry Pi. Slackware uses a "BSD style" init system, and there are other distributions (such as Gentoo, Arch, Crux, and MxLinux) offering OpenRC, Runit, and SysV. Each distribution has its own set of reasons for not defaulting to systemd, but here are 5 reasons I love the traditional BSD init system.

1. No Layers of Abstraction

Sometimes it's nice to have the exact details of a technology left abstract. It can be a real relief to not have to worry about how a process starts or stops and what command options are required for it to run in the foreground or background. When writing a systemd service script, you get a lot of functionality "for free." You don't have to write the case statement that detects the arguments provided with a command, you don't have to wrap a function in a script so it relaunches upon failure, and so on. You get a lot of functionality from the fact that systemd is designed to behave a specific way.

However, there are lots of good reasons you might prefer to write your own init scripts. Maybe you have your own conventions for starting and stopping and monitoring processes. Maybe you're using custom software that, for whatever reason, would benefit from more interaction types than `systemctl` has to offer. Or maybe you just enjoy writing and maintaining your own system scripts. There's no wrong reason, so if you want manual intervention past the level of defining that a service exists, then a traditional init system might be the way to go.

2. You Already Know How to Script

A systemd definition is a configuration file. There are key and value pairs, special options specific to the way systemd works, different definition types (service, socket, timer, and so on), and different locations for them on your system. It's invisible to most users, even if you're a user writing custom systemd configs, so you probably don't think about it much.

With many other Linux and BSD init systems, startup files are consolidated, singular in purpose, and relatively easy to understand. Assuming you already know how to write a shell script, you can construct a custom startup script using your favorite shell syntax and with identical logic to what you'd use for an interactive command. In fact, many init systems are just running startup scripts as shell scripts, and any time you want to start it manually, you can just execute the script yourself. You have a very direct relationship with a traditional init system, with the actual init application doing little aside from running a bunch of scripts for you upon boot.

3. Native Flexibility

Because an init script is just an executable script, you can actually write it in basically any language you like. A shell is safest because it's available early in the boot process and it doesn't usually have many dependencies. However, should you want

to try init scripts written in Python or Perl or even Awk, you can. I'm not actually encouraging you to experiment with your init sequence, but the flexibility is significant. You may have a good reason to write an init script in something other than `/bin/sh`, and a traditional init system can (probably) handle whatever you use.

Even if you very rationally keep your init scripts constrained to basic shell syntax, there's a lot of permissible flexibility within that script. You could use a case statement or an if statement to catch arguments. You could launch a subshell to gather important information from elsewhere on your system.

In short, you can use all the usual technologies available to you by your shell.

4. Just the Init Process

Critics of `systemd` often complain that `systemd` is “monolithic.” It's objectively not monolithic—you can look at the source code if you need proof—and yet somehow it does feel awfully monolithic. Because `systemd` has features for init, but also logging and scheduling and even user data management, it can seem like `systemd` is trying to do too much.

With a traditional init system, there's no confusion about what init is there for. Your executable init scripts are started at boot time and stopped at shutdown, and that's all. There are other

applications entirely to handle scheduled jobs and logs and whatever else.

5. Easy to Swap

Here's what I get when I run `sudo dnf remove systemd` on my laptop:

Problem: The operation would result in removing the following protected packages: `kernel-core`

It's hard to argue that removing an init system should ever result in the removal of the kernel, the single component that makes your computer functional. Even so, that's how `systemd` is very frequently packaged. This isn't actually a requirement of `systemd`, but a result of a package manager instructed to track dependencies.

It's rare, at least in my experience, to find a traditional init system with the kernel as a child package. That means it's often easier with a traditional init system to change which init system you use. Some distributions make it very easy to do this, with existing packages for different init systems for you to choose from. Others don't necessarily provide packages and scripts ready-to-use but allow you to easily swap out the init command for something different and to populate your init script directory with any converted scripts the different init system may require. I've run `minit`, `ninit`, `runit`, and `openrc` on

Slackware, with each one requiring a different amount of adaptation.

Choose Your Init System

Many users are happy to run the default init system used in their Linux or BSD operating system. In practice, that's what I do. I don't think too much about init until, of course, I need to interact with it. It's surprising how minor differences can impact your workflow, so having an init system you're comfortable and familiar with is a real luxury, especially in a pinch. However, if you've got a long weekend ahead of you with nothing better to do, it can be a lot of fun to try a new init system on a test computer. It may never matter, but then again you may just find an init system that you enjoy more than what you've been using. Whether you're switching from SysV to BSD, or BSD to systemd, or systemd to OpenRC, or whatever, exploring alternate ways of effectively accomplishing the same result is always educational.

5 Unix Books You Must Read

David Both
May 22, 2025

I am always learning new things. There are always new things to learn in the world of computers and technology. But there are also plenty of old things to learn, and that's what this article is about.

I have five books that I think are the most important books about practical computing and programming that have ever been written. That's not because they teach programming; they do not. But they teach more about programming in the Unix—and by extension, Linux—environments than all the other books I've ever read that are filled with code.

Together, they provide both history and context for the state of the computing world in 2025. All of them are relevant to Linux as much as they are to Unix. More than one provides some discussion of Linux and its place alongside Unix.

Together, these books have had a powerful and lasting impact on me as a SysAdmin, a programmer, and a consumer of computing power.

The Art of UNIX Programming

Eric S. Raymond, Sep 23, 2003

This is my favorite book about Unix development despite having very little actual code.

Raymond begins this book with a historical look at the philosophies of the early developers of the Unix operating system. He explores those philosophies—and his own—throughout the remainder of the book, using numerous case studies to illustrate good and bad examples of the art.

Each case is examined as an example of one or more of the philosophical principles and those principles and the realities of programming in a modern environment. One example is the idea of simplicity. More than one of the principles speaks to simplicity and the elegance of simple design. But Raymond and the Unix philosophies he's collected here recognize that sometimes a complex solution may be the only one, or at least the best.

Another important aspect of coding is that of data structures. More than one of the tenets covered in this book admonishes

the developer to design the data structures first. Elegant data structures facilitate simple, transparent, and elegant code.

The transcendent theme in this book is that of elegance, and that elegance is simple and beautiful. Elegant code is easy to understand and maintain

Raymond is clear about the fact that, while this is a book about Unix, all of the principles discussed apply to any system environment. He also points out that not all environments provide the tools necessary to implement all of the principles.

I like this book because it reminds me that programming is an art. Good code looks good, is easy to understand, and has an elegance that transcends clever coding. This is my favorite book because it has an incredible amount of development wisdom embedded in its pages. It's also one of the most readable books about Unix and development that I've ever read.

This book is available on Amazon and is freely available in its entirety at Linuxtopia [<https://www.linuxtopia.org/>].

Linux and the Unix Philosophy

Mike Gancarz, 2003

This is another favorite of mine. An extension of his earlier work, "The Unix Philosophy," Gancarz has included Linux in this edition.

Gancarz starts with the overarching Unix philosophy that its developers assumed that the users know what they're doing and shouldn't be shielded from any aspect of the operating system. This is the antithesis of Windows, which, descended directly from the DEC VMS operating system, is designed to protect and isolate users from the internal tools and full capabilities of the system.

Gancarz doesn't mention Windows, but because of their common ancestry, both DEC VMS and Windows usually provide only a single path to a computing solution; Unix and Linux commonly provide many paths. He also introduces Linus Torvalds and his "hobby," Linux, as a pivotal point in the history of computing by making a Unix operating system universal enough to be available to spread everywhere.

He outlines the deceptively simple nine primary tenets and ten minor tenets of his philosophy and then discusses them in detail in the rest of the book. For example, "*The Art of Unix Programming*," this book also hones in on the KISS principle, "Keep It Simple Stupid."

This book is the one that first prompted me to begin formulating my own philosophy, as I later laid out in my book, "*The Linux Philosophy for SysAdmins*."

This book is available on Amazon.

Unix—A History and a Memoir

Brian Kernighan

This is an interesting history of the creation and spread of Unix as told by one of its central figures. The book includes interesting anecdotes about how many of its most important features were created.

Although this book is interesting, it sometimes reads as a collection of reminiscences rather than a unified story. But I've read other memoirs with a similar structure so it's probably not uncommon.

The power of the stories in this book is the insight offered into the minds and activities of the original Unix developers. It describes a time in which computing resources, like CPU cycles, memory, and disk or tape storage, were all in very short supply. It's because of this that some of the tenets of the Unix philosophy were imposed on its architects. Simplicity and resource conservation were significantly more important than they are today. Fortunately, that doesn't prevent modern programmers from following the tenets of the philosophy.

Unfortunately, the e-book is available only as a poorly copied document in a PDF file, and it's restricted to use on devices that have the Kindle app or the Kindle itself. The hard copy is also available on Amazon, but I can't vouch for its readability because I haven't seen it.

The Cathedral and the Bazaar

Eric S. Raymond, 1997

Eric S. Raymond, an early proponent of open-source software, wrote this book to compare and contrast the two development models: the Cathedral and the Bazaar.

The cathedral represents the companies that employ tens or hundreds of programmers who create proprietary software based on a strict set of requirements that may or may not represent the true needs of customers. The resulting programs are sold or licensed under restrictive terms that preclude copying programs or using them on more than a single computer. In extreme cases, like Sun Microsystems, certain programs were licensed to run only on a specific number of processors, even if the computer contained more than that number; you could pay more money to be able to run the software on more processors.

By contrast, the bazaar confounded the proprietary world. This much less structured culture allows thousands and literally millions of programmers and testers to contribute to projects like the Linux kernel or the LibreOffice suite of office programs. Even more radical is that these open-source programs can be downloaded and used with no restrictions. Not only are they mostly free of charge, “free as in beer,” as they like to say, they are also “free as in speech,” because the license requires the source code to also be freely available for

anyone to download, view, and change if they have the skills. Open-source licenses also require that those changes be made available to the developers of the programs so they can be incorporated in future versions of the product if they work as advertised and are consistent with the goals of the project.

Raymond starts this book with another history, but of operating systems and software from a business viewpoint. Of course, the proprietary software companies couldn't—and still mostly can't—conceive of the fact that open-source software can be a viable business. Red Hat [<https://www.redhat.com/>] and Canonical [<https://ubuntu.com/>] have proven that it can.

The rest of the book concentrates on the struggle between the two models and their proponents while providing an interesting, if somewhat self-centered, view of the history of the open-source movement. For example, even though he discusses the origin of the term “open-source,” he doesn't attribute it to its creator, thus giving the erroneous impression that he coined it.

So, I think it's important to acknowledge that Christine Peterson, an American forecaster and the co-founder of the Foresight Institute, coined the term “open-source.” In February of 1998, Peterson was in a series of meetings with Eric S. Raymond, Jon “maddog” Hall, and many other leaders to discuss licensing Netscape as free software. A number of people, especially Peterson, did not think that “free software” properly defined what they were trying to accomplish. She came up with the term “open-source” and approached some of the other at-

tendees with the idea. In a meeting on February 5, some of the attendees started using “open-source” to describe software that was free of restrictions and for which the source code was readily available. In a comment at the end of the article on Opensource.com, Eric Raymond validates and supports Peterson’s account of the birth of this now-ubiquitous term.

Despite the narrow viewpoint, Raymond has given us an insightful look at the early software culture wars and the impact they continue to have today.

Available from Amazon.

Understanding the Digital World: What You Need to Know about Computers, the Internet, Privacy, and Security

Brian Kernighan

The new second edition of an important book by Brian W. Kernighan, one of the leading figures in the creation of Unix, author or co-author of many influential books, and professor of computer science at Princeton University. *Understanding the Digital World* combines computer history, technology, and personal stories, along with discussions about how today’s technology impacts our privacy.

Kernighan teaches a course at Princeton each year, “Computers in Our World,” intended for computer users who are not computer science majors. He wrote this book to bring much of the information contained in that course to the world at large.

Kernighan starts with an exploration of the technology itself. The title of chapter 1 is “What is a Computer?” Covering the CPU and how it works, he describes various forms of storage, including RAM, cache, disk, and other types of secondary storage, and how they all work together. After this overview of the hardware, he describes algorithms, how they are used to solve problems, and how they get incorporated into computer programs. In later chapters, Kernighan discusses the internet, the TCP/IP protocols that drive it, and some of the tools used to communicate using the internet. He also delves into the issue of security in a totally connected world.

My first inclination after reading this book was to send it to two of my grandkids that I helped build gaming computers for. This book is a good way for them to learn how computers work at a level they can understand. They can also learn about the pitfalls (beyond those their parents have discussed with them) of how to be safe on the internet. I also suggested to their parents that they read it, too.

It is not all gloom and doom. Far from it. Kernighan manages to scare me while simultaneously ensuring that readers understand how to mitigate the threats he discusses. In the vast ma-

majority of his scenarios, I had already implemented many of the protections he covers.

This book has made me think more closely about how I work and play on the internet, the methods I use to protect my home network, and how I use my portable devices. Kernighan's level of paranoia is sufficient to ensure that readers pay attention while reassuring us that we can still use the internet, our computers, and other devices with a reasonable amount of safety so long as we take the appropriate precautions.

No! I am not going to tell you all of that. You'll get no spoilers from me.

Overall this is a pretty easy read and accessible even for many non-technical readers. This was intentional on the author's part. So even if your technology quotient is fairly low, this book is still understandable. Despite the fact that he wrote the first edition of this book only a few years ago, this second edition includes important new material that makes it even more applicable to today's technology and the lightning-fast dissemination of data. I found the new section on artificial intelligence quite enlightening.

I highly recommend this book to anyone who wants to learn more about how computers work and impact privacy and security in the modern world.

Available from Amazon.

Although these books apply to other programming environments, the reality is that only Unix and its descendant, Linux, have the philosophy and the tools to implement them completely.

10 Reasons to Use Linux

David Both

August 26, 2025

Southern humorist, newspaper editor and columnist, comedian, and author Lewis Grizzard once said that there are two kinds of naked in the South. There's naked, where you don't have any clothes on, and there's nekkid, where you don't have any clothes on and you're up to something.

Well, I'm here to tell you the nekkid truth about Linux and it's most definitely up to something. It's up to helping people move away from Windows.

A new computer, whether desktop or laptop, costs a lot of money. Microsoft (M\$) is using the end of Windows 10 to try to force hundreds of millions of users to upgrade by purchasing new computers. Their response to people whose current computers won't support the arbitrary hardware requirements of Win11 is to throw away your old computer and purchase a new one with Win11 on it.

While that may be an excellent solution for both M\$ and the hardware manufacturers, it's a horrible option for you and the planet. It's based on the premise that old computers slow down, which is false. The computers run as fast as ever, but they're bogged down by the malware that collects on Windows computers. And the other marketing popups and junkware that just come standard with all new Windows computers are even more frustrating because they put them there intentionally. The companies that write the software being sold in those advertisements pay M\$ for the privilege of assaulting you while you're trying to work.

There's definitely a better option. Linux. Here are 10 excellent reasons to use Linux.

1. No New Hardware and No Licensing Costs

The good news is that Linux can replace Windows on your old computers and it costs nothing to download and install. Nada. Zero. Updates—that is, patches, fixes, and enhancements for the software that makes up the distribution—cost nothing. Upgrades from one release level to the next—such as from Fedora 42 to Fedora 43—cost nothing.

I know that's hard to believe and it seems counterintuitive. And it's not just Linux itself; it's all the software. In fact, I've only paid for one piece of software, \$49 US, in the 29 years I've

been using Linux and that was for a unique bit of software I wanted for testing.

Everything else I've used over the years has been free as in beer. That includes all of the following and more.

- LibreOffice, which I use for spreadsheets, presentations, and writing all 9 of my books.
- Firefox for web browsing.
- Thunderbird for email, although I use Alpine—a retro, text-mode email client—more. Yeah, that's the geek coming out.
- Audacious to play my music CDs and the music files I've copied from them.
- Audacity for recording and editing audio files.
- GNUCash for my personal and business accounting.
- Okular for viewing and marking up PDF and other document types.

And all of the software necessary to run this website, including these and more.

- Apache web server to deliver the web pages.
- WordPress for content management.
- MariaDB (MySQL) for data storage and management.

2. Linux Easily Runs on Really Old Computers

Did you know that Linux can revitalize old computers and keep them running for decades?

Those older computers, like my 20-year-old Dell Optiplex GX620, can run the most modern versions of Linux. Mine's been running Linux since it was given to me a few years ago and is now on Fedora 42, which was released just this April. Yes—it's slower than my newer computers, which are between 2 and 12 years old—but that's only because its 2-core Pentium 4 processor and 4GB of memory are by their very nature slower than newer computers with more and faster CPUs and memory, like my home-built system with 20 cores, 28 CPUs, and 32GB of RAM. But they're both running at their designed speeds.

So long as you can replace the internal parts when they break, you can keep old computers running indefinitely.

Yes, it's in my garage.



My 20-year-old Dell Optiplex GX620. Image by David Both, CC-by-SA 4

3. Enhanced Privacy

Windows comes with lots of ads and spyware. This slows down your computer, lets companies spy on you, and increases your energy bills. It's also annoying when those ads pop up to interrupt you in the middle of your work, or worse, a presentation.

Windows also requires you to create an M\$ account just to install it, and that requires some personal information. Information that I certainly don't want to reveal to any more organizations than necessary—unless it's by my own choice.

Linux doesn't upload images of my desktop or any of my files and documents to the M\$ cloud “for security.” Linux allows you to keep your data private.

4. It's Good For the Planet

The production of a computer accounts for 75%+ of carbon emissions over its lifecycle. Keeping a functioning device longer is a hugely effective way to reduce emissions. With a Linux operating system you can use your device longer.

5. Community & Professional Support

There are local repair cafes and independent, professional services and computer shops available that can provide you help. You can find support in online forums, too.

Linux Mint has excellent online documentation [<https://www.linuxmint.com/documentation.php>] that includes an installation guide and a user guide. You can read them online or download them in PDF or ePub formats. Free of charge, of course.

Fedora also has excellent online documentation [<https://docs.fedoraproject.org/>] that includes an overview and links to documentation about the different Fedora options, like Fedora Server and Fedora Workstation. There are also links to the many spin and lab releases like KDE Plasma, Xfce, Python Classroom, Astronomy, Games, and more.

6. Better User Control

Linux grants you the four freedoms of software. You are free to use, study, share, and improve the program for as long as you wish. You are in control of your device, your software, and your data.

7. Familiar Interface

It might surprise you to know that Linux can look and work much like Windows. Cinnamon is the default desktop for Linux Mint because it's been proven to be an excellent choice for users who are migrating from Windows. Cinnamon is a modern, innovative, and full-featured desktop while also offering a secure, familiar feeling to Windows users.

8. Choices

One of the reasons I like Linux is that it offers so many choices for everything. From the desktop and the distribution to the open source software you install and use, there are always multiple choices from which to choose. The color scheme, desktop background, icon scheme, and more are all choices you can make to personalize your desktop working environment.

Some people, like me, like having all those choices. Others not so much. But one of the choices you have is to make no choices and leave it just as it is.

9. Best Desktops for New Users

With all those choices, the most important one you can make is that of the desktop. The distribution is also important, but the desktop will define how you view and interact with your computer as well as the amount of flexibility you have to personalize your experience.

Linux has a large number of desktop interfaces, but there are only three that I think are best for users new to Linux

- Cinnamon—The best option for new Linux users. It's modern, innovative, and has some features found in more complex desktops while not overwhelming new users with customization options. I find it's great for most systems, but it crashes on systems with only 4GB of RAM, like my old Dell.
- Xfce—A lightweight desktop with plenty of configuration options but not so many as to be overwhelming. It has a clean look and is very responsive even on that old Dell.
- KDE Plasma 6—A powerful, complex desktop with more personalization options than I've seen in any other desktop. But it is perfect for newcomers without making any changes while offering those options for when you're ready for them.

I personally think Cinnamon is best for most new users. However, you can install multiple desktops and switch between

them as you want to. I change desktops frequently as a way to keep my experiences fresh and to see what new releases bring.

Pick your desktop first, then the distribution.

10. Best Distributions for New Users

Some distributions and desktops are better for users new to Linux. I prefer Fedora [<https://fedoraproject.org/>] and it's highly recommended by many for use by new Linux users. Others prefer Linux Mint [<https://www.linuxmint.com/>] and I think that it's a great choice as well. You can't go wrong with either of these distributions and they both offer all three of those desktops.

Try Linux; it's easier than you think.

A Brief History of the Linux Desktop

Jim Hall

May 19, 2025

I first discovered Linux in 1993 when I was an undergraduate student. I “grew up” on DOS, and the Unix systems in our campus computer lab were both familiar (because of the command line interface) and more powerful (multitasking and more tools). I wanted something similar for my computer at home and eventually found this “Linux” operating system, which was just like the “Big Unix” systems on campus, but it included source code.

In the early days, people often copied a working Linux system from someone else. In 1992, Softlanding Linux System (SLS) was the first Linux *distribution*. It installed the operating system as *packages* so you could choose what packages and package groups you wanted to install. I installed SLS 1.03 in May 1993.

Back then, Linux needed only 2 MB of memory to boot up and use the command line, or 4 MB if you wanted to compile programs. If you wanted to run the X Window System, you needed 8 MB of memory. That meant my '386 computer had *just* enough memory to run Linux with X.

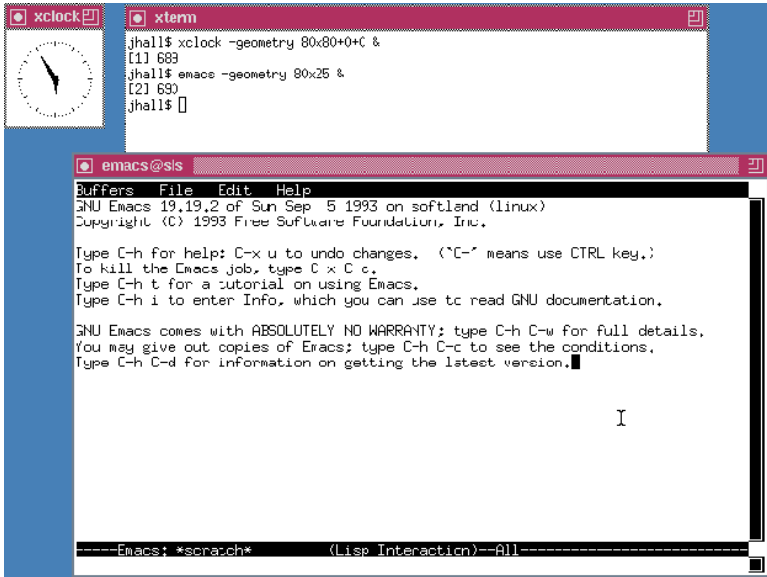
Like many people in the 1990s, I ran Linux in a “dual boot” configuration, so I could use a boot menu to selectively start up my computer with either Linux or DOS (or later, Windows 95). By the time Windows 98 came out in 1998, I didn't need Windows anymore, and I went “all in” with Linux.

Since that first experiment with Linux, I have used Linux on my desktop. That's 32 years since I first installed Linux and 27 years running Linux exclusively. I've watched Linux “grow up” over the years, especially with the graphical desktops. Let's look back on a *brief history* of the Linux desktop:

X and Window Managers

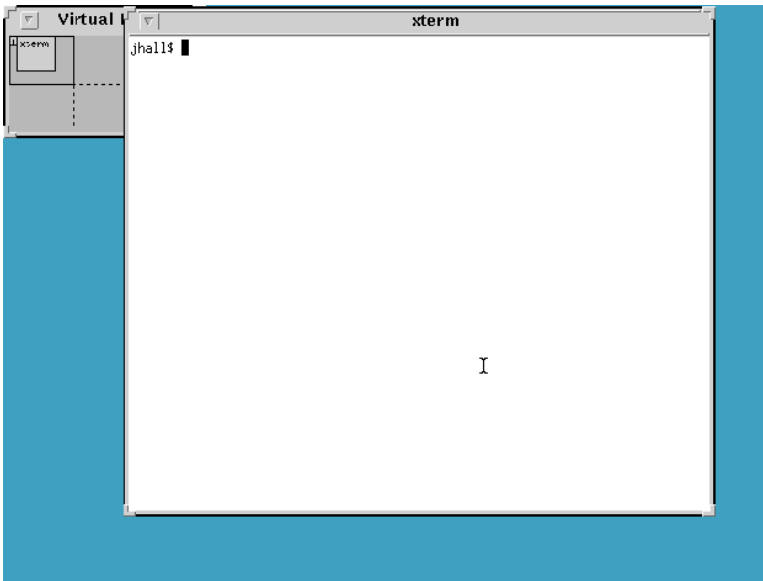
The first graphical environments for Linux weren't really *desktop* systems. Instead, you ran the X Window System with a *window manager* on top of it. X provided the basic components for a graphical user interface, such as using the mouse, and the window manager actually *managed* the different program windows that ran inside X.

The *de facto* window manager in 1993 was TWM, which dates back to 1988. TWM was quite simple and provided basic window management, but it did the job and matched the TWM system running on the SunOS systems in our campus computer lab:



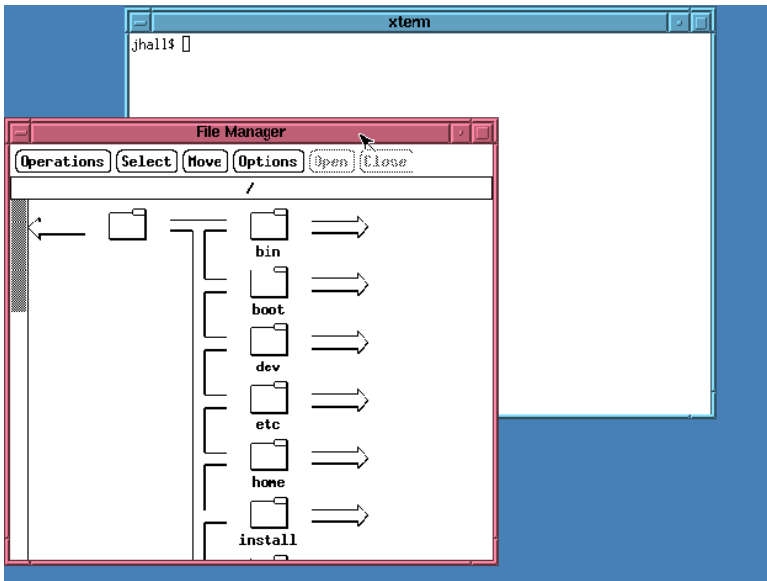
TWM running on SLS 1.05, showing xterm, xclock, and emacs

Another early window manager was the Open Look Virtual Window Manager (OLVWM). Developed by Sun Microsystems, Open Look was also ported to other Unix-like systems. And as a *virtual* window manager, OLVWM supported multiple *workspaces*. That meant you could use one workspace for your terminal and editor so you could write and compile programs, another workspace for your email, and so on.



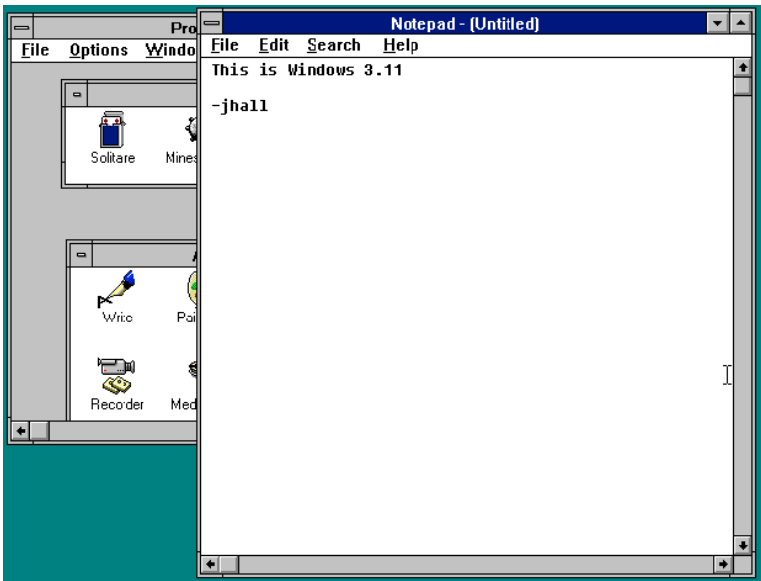
OLVMW running on SLS 1.05, showing xterm and the workspace selector

As Linux grew in popularity, others created new window managers that improved the “desktop” experience. One of these early Linux window managers was FVWM, which also supported virtual workspaces. FVWM also had a more modern look than TWM or OLVMW and initially looked like the Common Desktop Environment with the Motif widgets from “Big Unix” systems:



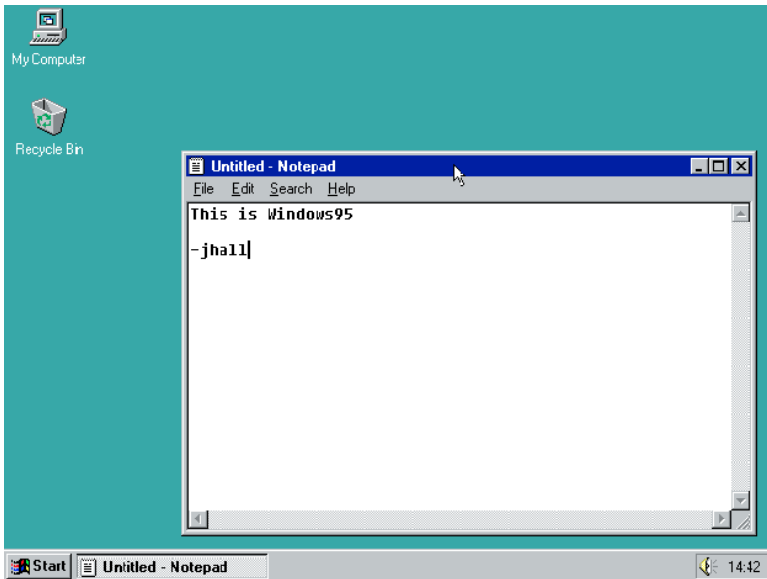
FVWM running on SLS 1.05, showing xterm and a file manager

Modern Linux users might think that these early desktops are quite plain, but it's important to remember they were similar to other graphical environments of the day. Until 1995, Windows had a similar interface. Windows used simple window controls and used a *program manager* to launch new applications:



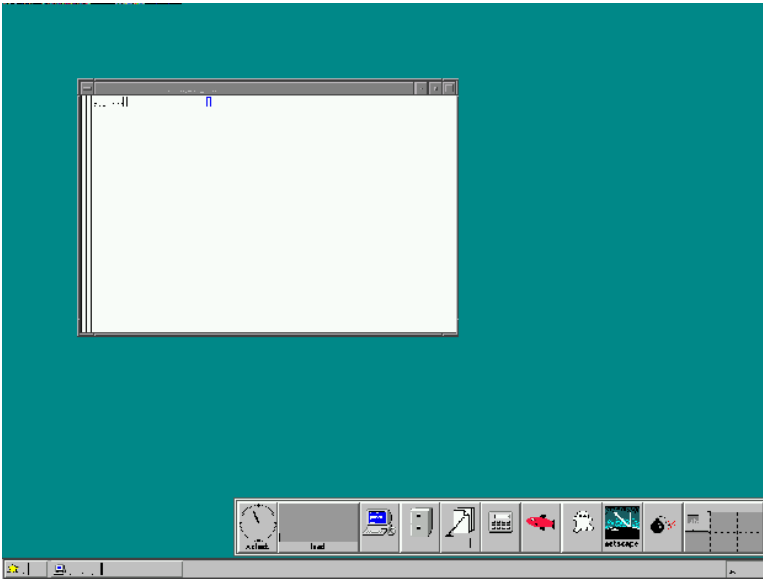
Windows 3.11 showing the Program Manager and Notepad editor

In August 1995, Microsoft released Windows 95 and changed the desktop interface, at least on PC. I didn't like Windows 3, but I thought Windows 95 was okay. Most importantly, Windows 95 was a proper *desktop* system, and you could even put *icons on the desktop*. That seems like a small thing now, but it was a big deal at the time.



Windows 95 showing the desktop and Notepad editor

The Windows 95 interface was much easier to use than previous versions of Windows and even other Linux window managers. Not to be outdone, Linux developers updated the FVWM window manager to mimic the Windows 95 interface. FVWM95 wasn't a *desktop* but it looked like one, including a version of the "Start" menu and a taskbar that showed the programs you were running.



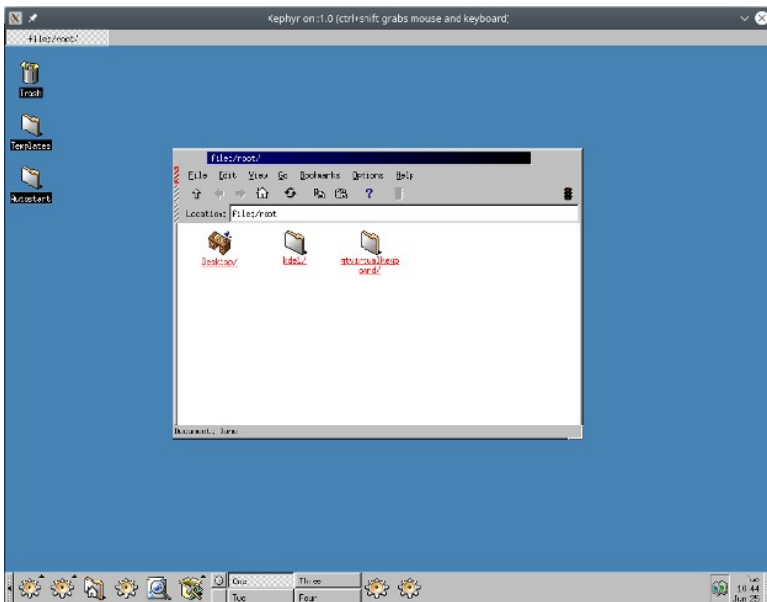
My version of FVWM95 has some graphical hiccups now, but it worked fine on the original Red Hat Linux 5.2

FVWM95 and other window managers were improving, but none were really an integrated desktop. Linux applications that used a graphical user interface under X might use different “graphical toolkits,” so they might look and operate completely differently from each other. And you couldn’t copy and paste between applications, except with the plain text “copy/paste” feature provided by the underlying X Window System. What Linux really needed was a do-over with an integrated desktop environment.

The First Linux Desktop

In 1996, Matthias Ettrich wanted to have an easy-to-use desktop for Linux. And more importantly, he wanted to make everything *integrated* like a proper desktop environment.

And that's how the K Desktop Environment got started. The K is for "Kool," but the "KDE" name was also meant to be a play on the Common Desktop Environment (CDE) name. Finalized in July 1998, KDE 1.0 was a big step forward for Linux on the desktop:



K Desktop Environment (KDE) version 1.0

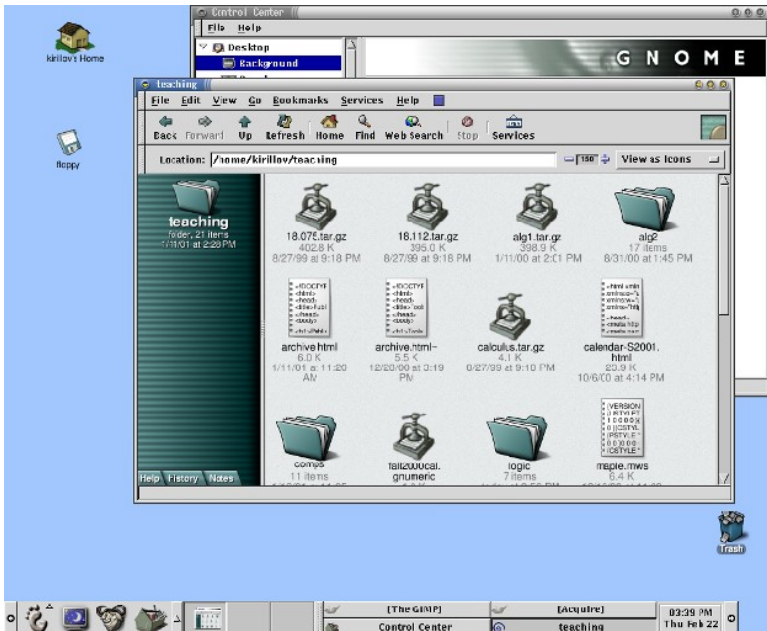
Finally, Linux had a true desktop with application integration and more modern desktop icons. KDE's design was not dissimilar from Windows 95. You had a kind of taskbar along the

bottom of the screen that provided the equivalent of Windows 95's Start menu as well as several application shortcuts. KDE also supported virtual desktops, which were cleverly labeled One, Two, Three, and Four. Running applications were represented via buttons in a separate taskbar at the top of the screen.

Unfortunately, not everyone was happy with KDE. To abstract the GUI from the system, KDE used Trolltech's Qt toolkit library. Trolltech allowed Qt to be used at no charge in free software applications but charged a fee to use it in commercial or proprietary applications. And that was not aligned with the Free Software movement. This caused problems for some Linux distributions: should they include KDE or default to an older but all-free-software graphical interface?

In response, Miguel de Icaza and Federico Mena started work in 1997 on a new Linux desktop. The new project was dubbed GNOME, for GNU Network Object Model Environment. GNOME aimed to be completely Free Software and used a different toolkit, called GTK, from the GIMP image editor. GTK literally stood for GIMP Toolkit. When GNOME 1.0 was finally released in 1999, Linux had another modern desktop environment.

Introduction to Linux



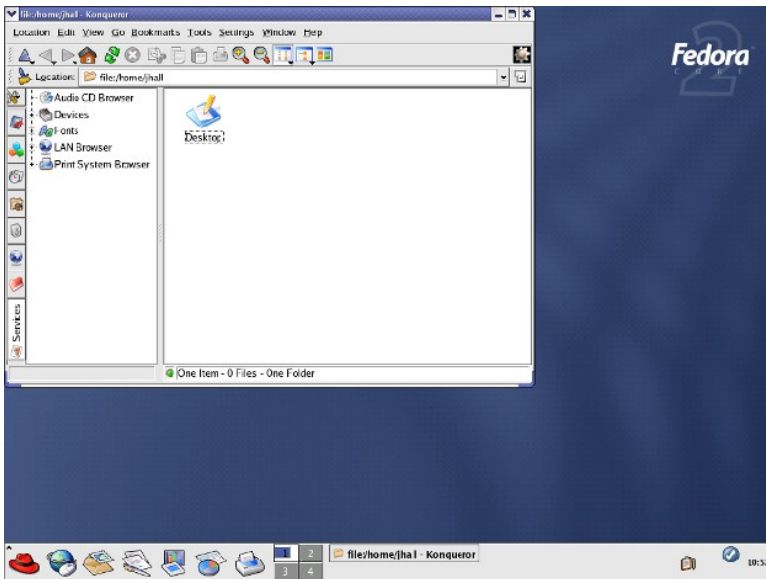
GNOME version 1.0

The “KDE versus GNOME” rivalry continued for some time, focused around Free Software issues with the Qt license. By 1999, Trolltech re-released the Qt library under a new public license, the Q Public License (QPL). But the new license carried its own baggage: the QPL only applied to Qt’s use in open-source software projects, not commercial projects. This meant the QPL was not compatible with the GNU General Public License (GNU GPL). This licensing issue would remain until Trolltech re-re-released the Qt library under the GNU GPL version 2 in 2000.

The Linux Desktop Grows Up

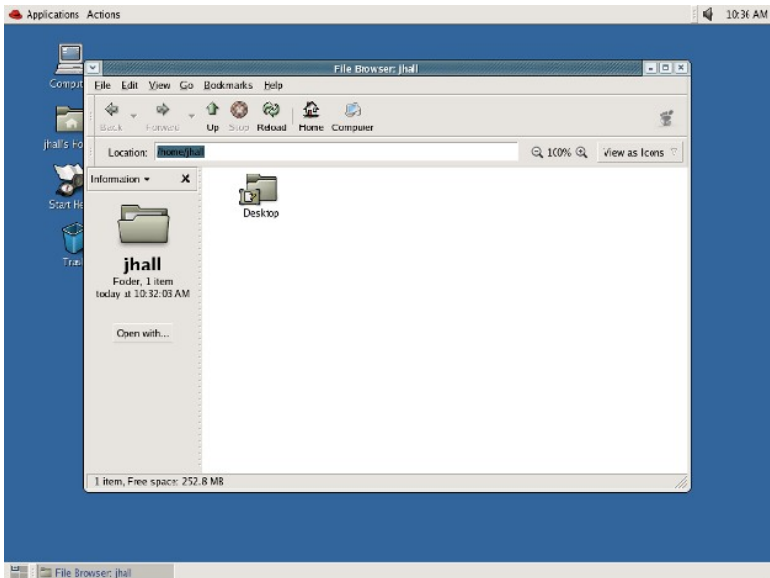
After that, the Linux desktop continued to mature. And while other desktop environments appeared, GNOME and KDE got most of the attention.

Over time, both KDE and GNOME had made significant strides yet brought only incremental changes to the user interface. For example, KDE 2 and 3 continued to rely on a taskbar concept at the bottom of the screen but incorporated the buttons for running applications. One of KDE's most visible changes was the addition of the Konqueror browser, which first appeared in KDE 2.



KDE 3.2.2 (2004) on Fedora Core 2 showing the Konqueror file manager, using a Fedora Core 2 theme

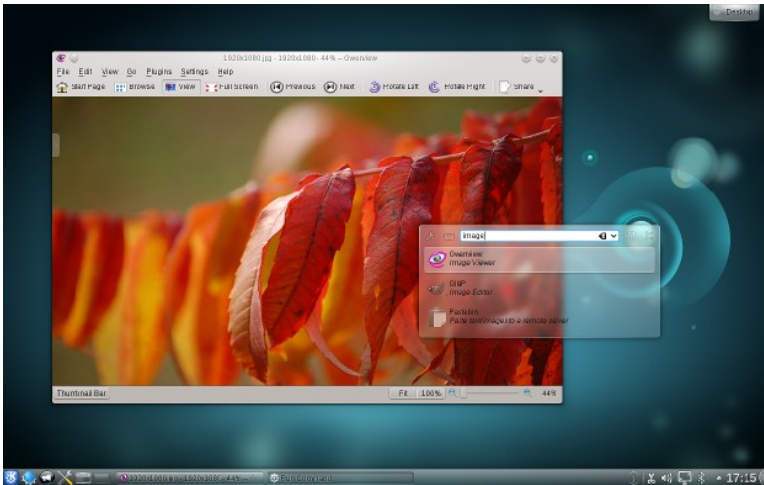
GNOME 2 also used a taskbar concept but split the bar into two: a taskbar at the top of the screen to launch applications and respond to desktop alerts, and a taskbar at the bottom of the page to show running applications. I liked to refer to the two taskbars as “things you can do” (top) and “things you are doing” (bottom). In addition to the streamlined user interface, GNOME also added an updated file manager called Nautilus, developed by Eazel.



GNOME 2.6.0 (2004) on Fedora Core 2 showing the Nautilus file manager, using a Fedora Core 2 theme

Over time, KDE and GNOME have taken different paths. Both provide a feature-rich, robust, and modern desktop environment, but with different user interface goals. KDE 4.6 (January 2011) and KDE 4.7 (July 2011) provided a more traditional

desktop metaphor and continued to rely on the taskbar concept. But GNOME 3 (2011) completely changed gears with a new desktop concept. GNOME 3 aimed to create a simpler, more streamlined desktop experience, removing the taskbar entirely in favor of an “Activities” overview that showed all open applications as separate windows.



KDE 4.6 showing the Gwenview image viewer

Linux on the Desktop

Having your choice of desktop environments means users can pick the environment that meets their needs. While Linux has other desktop systems to use, the two most popular Linux desktops today are KDE and GNOME. Their current versions are both very mature and packed with features. Both KDE and

GNOME try to simplify and streamline the Linux desktop experience, but in different ways.

GNOME continues to aim for a minimal appearance, removing all distracting user interface elements so users can focus on their work. KDE takes a more familiar approach with the taskbar, adding other visual improvements around improved widget handling and icons.

At the same time, you don't completely lose out on compatibility. Every major Linux distribution provides compatibility libraries, so you can run applications from, say, KDE while running GNOME. This is immensely useful when an application you really want to use is written for the other desktop environment.

I don't see this changing anytime soon. And I think that's a good thing. No matter what desktop you use, you have a modern desktop with great integration. And above all, this means Linux has the best feature in free software: choice.

Image credits:

- GNOME 1.0—*Red Hat Software*
- GNOME 3.0—*with permission from GNOME (2019)*
- KDE 1.0, KDE 2.2.2, KDE 4.6—*with permission from KDE (2019)*

A Fresh Look at Manjaro KDE Plasma

Don Watkins

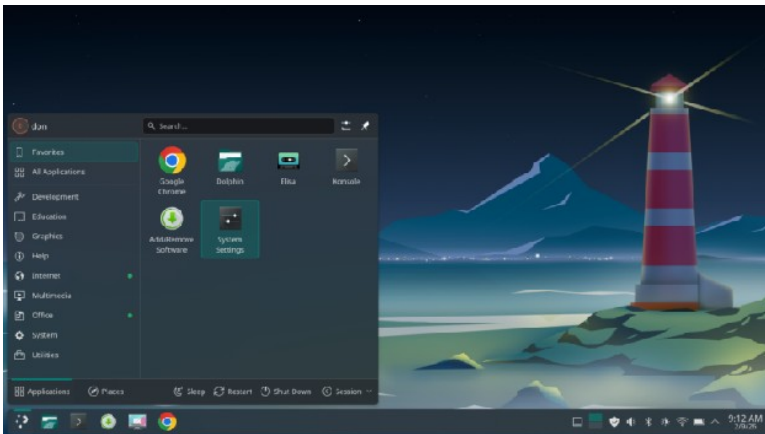
February 10, 2026

Linux adventures come in many distributions, and Manjaro is no exception. I decided to download the KDE Plasma version of Manjaro and install it on my Latitude 7410 laptop. Last year, I installed the Fedora version of KDE Plasma and really enjoyed the experience. I got used to using KDE tools like Spectacle, which are invaluable for screenshots when writing articles. I liked the desktop interface, which is similar to Cinnamon in many respects but offers a more granular approach to settings and is more intuitive.

My familiarity with Plasma encouraged me to try out the Plasma Manjaro spin. I downloaded it on my Linux Mint Cinnamon desktop and created a USB drive for the installation on my laptop. Once that was complete, I was ready to begin the installation. I inserted the USB drive into the Latitude laptop and pressed F12 during startup to select the USB boot option.

Soon, I was greeted with the live Manjaro KDE Plasma desktop.

I selected the installation option, and soon I was presented with several choices, including the file system type to use. Unlike Fedora and some other distributions, Manjaro allows you to choose between Btrfs (the default), EXT4, XFS, or F2FS. I opted for EXT4 because I am familiar with it. Many Linux users appreciate having this choice. Linux has always been about choice, and Manjaro lets you build your system your way. This philosophy underpins the distribution, which is based on Arch Linux. While Arch is suited for a do-it-yourself approach, Manjaro makes those choices more accessible to everyone. The installation completed quickly, and soon I had a beautiful Plasma desktop. Manjaro offers a choice of office productivity software to install. I chose LibreOffice.



Screenshot by Don Watkins CC-by-SA 4.0

Manjaro's Pamac [<https://github.com/manjaro/pamac>] software center offered me many options to customize my workflow. I started by downloading and installing the Chrome browser, which is still my favorite. Manjaro KDE Plasma comes with Firefox as its default web browser. I quickly opened Pamac and enabled Flatpaks, a simple process that takes only a couple of mouse clicks. After that, I swiftly installed Joplin and Visual Studio Code (VS Code). If you're an experienced Plasma user, you will feel right at home in this implementation.

Manjaro is a rolling release [https://en.wikipedia.org/wiki/Rolling_release] distribution, which offers a new experience for me. It is a suitable choice for both beginners and experts. Manjaro aims to bridge the gap between the power of Arch Linux and the needs of everyday users. Its goal is to make the strengths of Arch—such as rolling releases, the Arch User Repository (AUR), and the pacman package manager—accessible without requiring users to build their systems from scratch. Manjaro achieves this with the Plasma desktop environment, one of the most customizable Linux desktop environments. Despite its features, it remains surprisingly lightweight, using only slightly more memory than Xfce. Additionally, it comes with many built-in tools, such as Dolphin File Manager [<https://apps.kde.org/dolphin/>], KRunner [<https://userbase.kde.org/Plasma/Krunner>], and Spectacle [<https://apps.kde.org/spectacle/>]. Overall, it runs smoothly on my four-year-old Latitude 7410.

My time with Manjaro KDE Plasma has reminded me why exploring new Linux distributions is so rewarding. Manjaro takes the power and flexibility of Arch and wraps it in an experience that feels welcoming, polished, and thoughtfully designed. The Plasma desktop only enhances that impression; its balance of customization, performance, and intuitive design makes it a natural fit for users who want a system that adapts to them rather than the other way around.

Running on my Latitude 7410, Manjaro Plasma has been fast, stable, and enjoyable to use. The combination of Pacman, Flatpak support, and the AUR opens the door to an enormous software ecosystem, while the rolling-release model ensures the system stays fresh without periodic reinstallations. For someone like me—comfortable with Linux but still curious about what lies beyond the familiar Debian- and Fedora-based worlds—Manjaro offers a compelling middle path.

Whether you're a seasoned Plasma user or simply looking for a distribution that blends Arch's strengths with a more approachable experience, Manjaro KDE Plasma is well worth a look. It's a reminder that in the Linux world, there's always something new to discover, and sometimes, the best surprises come from stepping just a little outside your comfort zone.

Best Linux Distros for Windows Users: From Mint to Pop!_OS

Don Watkins

October 2, 2025

When you are new to Linux, it can be challenging to navigate all the options available to users. Choosing a distribution can be difficult, especially considering all the people trying to convince you that their favorite should also be yours. Here's a quick look at the top 5 that we at Both.org recommend for Linux newcomers.

1. Linux Mint Cinnamon

If you're migrating from Windows 10 or Windows 11, I recommend Linux Mint Cinnamon. This is particularly true if you have a computer with an Intel i5 or Ryzen 5 processor or higher and more than 4 gigabytes of RAM. The Cinnamon

desktop resembles a Windows desktop, and its interface will look and feel very familiar.



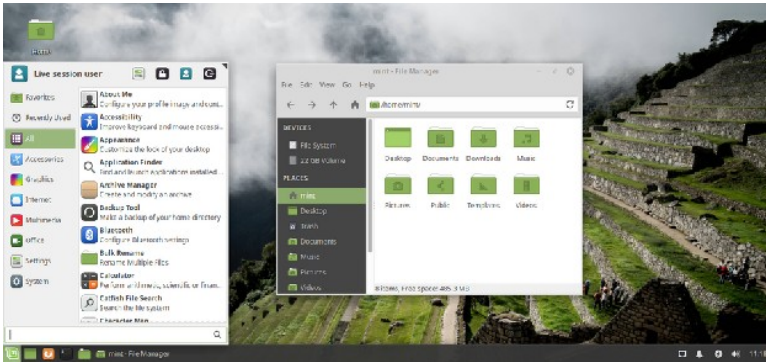
Screenshot by David Both. CC by SA 4.0.

The LM button to the far left of the taskbar is in the same spot as the Start menu on Windows. The interface is overall intuitive, and it's easy to navigate. Linux Mint has excellent documentation [<https://linuxmint.com/documentation.php>] to support new users. The documentation explains how to download the iso file and create a bootable image and has a very complete installation guide.

2. Linux Mint Xfce

However, you may be a person who purchased a lower-powered computer that has fewer than four gigabytes of RAM and

an i3 or lesser processor. You should consider downloading and installing Linux Mint Xfce [<https://linuxmint.com/edition.php?id=323>]. It has excellent documentation [<https://linuxmint.com/documentation.php>] and a user guide that is shared across all Linux Mint distributions featuring a nearly identical interface. A button at the bottom left of the display serves as the launch menu for all installed applications on your system.

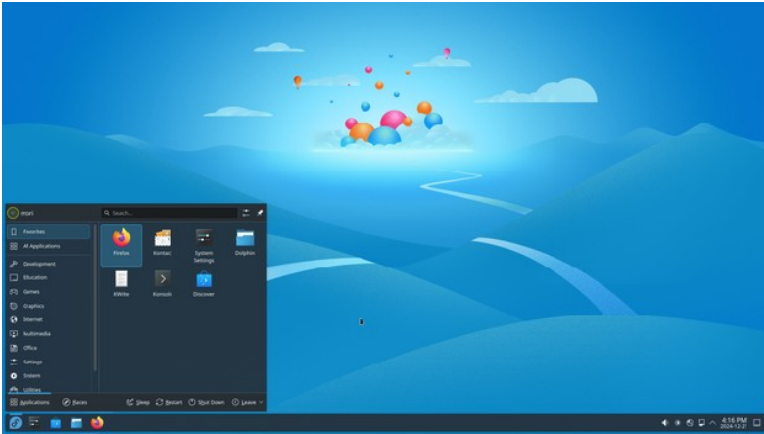


This file is licensed under the Creative Commons Attribution-Share Alike 4.0 International license.

3. Fedora KDE Plasma

If Linux Mint is not what you had in mind for a Linux desktop experience, you may want to consider Fedora KDE Plasma. It works best on an i5/Ryzen 5 or higher with at least 4 GB of RAM. It's easy to download and install. Use tools like Rufus (Windows) or Balena Etcher (Linux/macOS). Once you have a

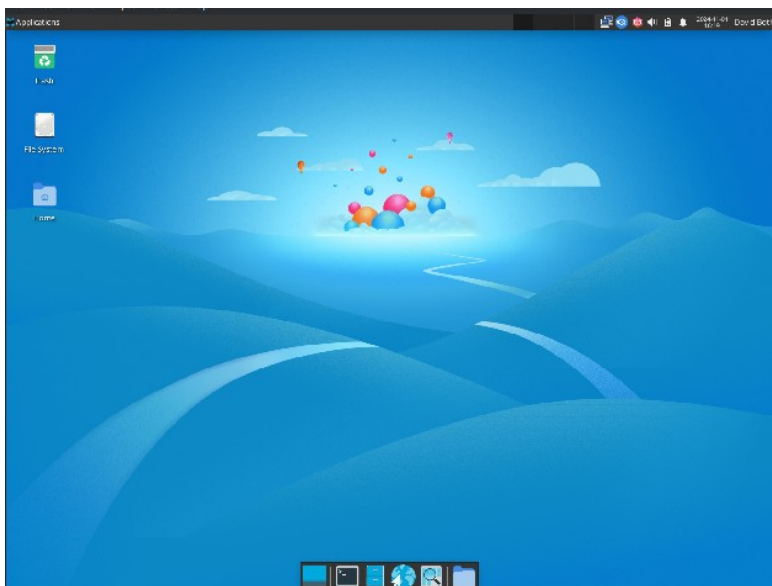
bootable USB drive, installation will vary slightly depending on how your particular computer system allows you to boot from a USB drive. Fedora has excellent documentation for the KDE Plasma desktop. The Fedora KDE Plasma desktop employs a desktop metaphor similar to that used by Windows users.



This work is free software, distributed under the GNU Public License from Wikimedia Commons

4. Fedora Xfce

If your computer has an older or lower-powered processor and less than two gigabytes of RAM, you should consider Fedora Xfce [<https://fedoraproject.org/spins/xfce/>]. You can download Fedora Xfce and create the installation media in the same way as described above.



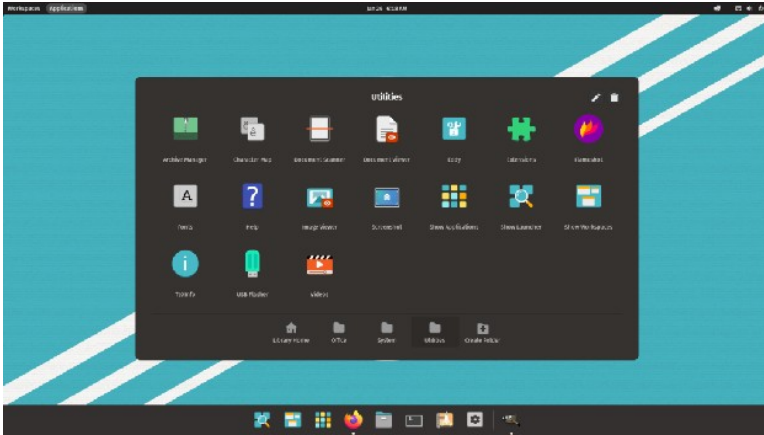
Screenshot by David Both. CC by SA 4.0.

Fedora Xfce offers a desktop paradigm similar to it, so it's familiar to Windows users, ensuring a pleasant desktop experience.

5. Pop!_OS

Suppose you have the power and are willing to think a bit differently than the traditional Windows desktop paradigm. In that case, you owe it to yourself to try System76's Pop!_OS [<https://system76.com/pop/>], which is derived from Ubuntu but, unlike the former, supports Flatpak out of the box. Sys-

tem76 has excellent documentation [<https://system76.com/support/>] to support Pop!_OS users.



Pop!_OS desktop

Pop!_OS is built for modern Intel/AMD processors with at least four gigabytes of RAM but recommends at least eight gigabytes. According to the System76 website, Pop!_OS is designed “to encourage discovery, Pop!_OS is designed for fast navigation, easy workspace organization, and fluid, convenient workflow.” It just gets out of your way. Pop!_OS is under continuous development by the System76 team. It is a customized GNOME [<https://www.gnome.org/>] desktop environment.

Pop!_OS offers seamless out-of-the-box compatibility with both AMD and Nvidia graphics cards. It comes preconfigured with disk encryption for enhanced security, intuitive window and workspace controls, and efficient keyboard shortcuts for smooth navigation. Built-in power management profiles help

optimize performance and battery life. Recent versions also include packages that simplify the setup of TensorFlow and CUDA, making it a strong choice for developers and AI enthusiasts.

Breaking Free from Windows 10: KDE Plasma Is Your Upgrade to Freedom

Don Watkins

June 17, 2025

October 14, 2025, is a big day if you're a Microsoft Windows user. That's the day that support for Windows 10 ends. But what if you're happy with the computer you have now and don't feel like spending more money to stay with the guys from Redmond? Is there a choice for you? You bet. The KDE Community has precisely what you need. It's KDE for Windows 10 Exiles [<https://kde.org/for/w10-exiles/>]. You now have a choice, and all you need to do is upgrade your operating system.

That might seem like a tall order to someone who has never done that sort of thing, but it's not that difficult. The KDE Project has a website dedicated to helping you do precisely that. KDE recommends trying the Plasma desktop. You'll be glad you did, too. That means you get to keep the computer you already have, and it means that from here on out, you'll be

in the driver's seat, along with the rest of us, using free and open-source software. KDE's Plasma desktop ensures that "you get all the advantages of the safety, stability, and hi-tech of Linux, with all the features of a beautiful, modern, and powerful graphic environment."

There are several good reasons to choose Plasma, and one of them is that it's easy. The Plasma desktop doesn't look that different from the one you have been using. The launch menu is in the bottom left-hand corner, just like it is on Windows 10. Plasma is powerful and lets you move things around so that your workflow could be precisely what you need it to be. The best thing about Plasma is that it's free. It can be on your computer or everyone in your department could upgrade to Plasma for free. Plasma is powerful. You'll discover that Plasma lets you have up to four desktops. You're no longer constrained to just one.

Organizing your files and folders requires some planning, but it's an easy task. Get a USB-compatible external hard drive and store all the data you want to keep on it. Later, Plasma will have no trouble reading it. Most of the hardware you'll use, whether it's WiFi or Bluetooth, will be configured by the operating system. If you're a Microsoft Office user, your documents can be read by LibreOffice, which is included in most standard Linux distributions.

Moving to Linux means that you'll be giving up viruses and virus-related scams. You won't be getting any advertisements

from the folks in Redmond. There won't be telemetry built into your operating system and its software. You will begin to enjoy absolute privacy, and you won't be forced to upgrade. Your computer will last much longer. There are no hidden costs with Linux.

I am a long-time Linux user, but I've recently converted to KDE Plasma on Fedora 42 [<https://fedoraproject.org/kde/download/>]. Many Linux distributions feature the Plasma desktop [<https://kde.org/distributions/>]. There are many other desktop options for Linux, so if you don't care for Plasma, be sure to check these other desktop environments: GNOME [<https://www.gnome.org/>], Xfce [<https://xfce.org/>], LXQT [<https://lxqt-project.org/>], MATE [<https://mate-desktop.com/>], and Cinnamon [<https://projects.linuxmint.com/cinnamon/>].

Choose your new computer's operating system

Alan Formy-Duval

October 8, 2025

I recently built a new computer, and now it needs a fresh operating system. But before I install it, I want to explore my operating system options. This exploration can be time-consuming since I need to download, install, and run each prospective OS. This starts with downloading the installation media in the form of an ISO file. Each ISO file is then written to a USB flash drive to create the bootable installer. Most of them include a convenient “LIVE” option for testing the OS. This means it doesn’t alter anything on the target computer when I first boot it. Once I decide which one I like, it can be installed by simply clicking an icon for full installation. It is possible to try an operating system in a virtual machine, but this doesn’t provide an opportunity to test against specific real hardware.

Test Boot

I like to research hardware and driver support before I purchase. However, the only way to guarantee hardware compatibility is to run it. One example is that with my new computer, I want to use S/PDIF digital optical audio input/output. I was fortunate that the sound worked great for me with all Linux distributions that I tried.

Ventoy

Fortunately, for full hardware trials, we can expedite the process with a multi-boot manager. Ventoy is one such utility software that formats a flash drive with a multiboot manager and loader. It offers a menu of OS options to boot from based on the ISO files that it finds in its root partition. You still need to download the ISO files and copy them to the flash drive, but now they can each be booted by selecting them from the menu and running in “LIVE” mode before being installed. This can be done on a computer that already has an OS set up or one that doesn’t even have physical drives installed yet. It is fairly easy to install and use and you don’t need to edit any configuration files. Simply drop the ISO files onto the root of the flash drive and Ventoy will detect them.

Ventoy allows me to try out many operating systems on my new hardware in a short period of time.

Distrowatch

We have a brand new computer, a Ventoy flash drive; “it’s dark, and we’re wearing sunglasses.” A great place to discover different operating systems is *distrowatch.com*.

“DistroWatch is a website dedicated to talking about, reviewing, and keeping up to date with open-source operating systems.” – [Distrowatch].

I have been using it since it began in 2001. So, that’s where I’m heading next: “Hit it!” There you will find a comprehensive listing of operating systems. You can search by various criteria, such as type, country of origin, or the desktop environment they include. Distrowatch also maintains usage rankings so you can get some idea of what is popular and what might be fading out. Links to distributions’ main pages are provided to make it easy to browse and download their ISO image files.

Findings

My new 2025 system—consisting of an AMD B850 chipset and Ryzen 5 CPU—worked great with all distributions that I tried. Ventoy also worked great, allowing me to boot up Fedora, Ubuntu, Pop!_OS, and others in a matter of minutes. Most of the differences are purely aesthetic and related to the settings offered and the desktop or window manager installed by default. My final decision was to continue using Fedora Linux

Workstation, which I have been using for several years. I also checked out the latest beta of Pop!_OS since that is what my daughter uses. This way I have some idea of what to expect when she is ready to upgrade.

Once I made my decision, I bolted in the NVME boot drive and installed my chosen Linux distribution – it's just that simple. Give it a try!

Have fun!

Cinnamon: A Tale of Forks and Familiarity in the Linux Desktop Landscape

Don Watkins

January 13, 2024

As a long-time Linux user, I've witnessed firsthand the power of a familiar interface in welcoming newcomers to open-source computing. The transition from Windows or macOS can be daunting, navigating unfamiliar layouts and paradigms. This is where Cinnamon shines. Its unwavering commitment to the traditional desktop metaphor—the panel, menus, and applets we know and love—provides a comfortable bridge, easing users into the vast potential of Linux without sacrificing ease of use.

I have used Cinnamon across prominent distributions like Fedora [<https://fedoraproject.org/spins/cinnamon/>], Ubuntu [<https://ubuntucinnamon.org/>], and its home, Linux Mint [<https://linuxmint.com/>]. While all of the implementations are good, Linux Mint is my favorite because it supports Flatpak [<https://flathub.org/en>].

Within the vibrant ecosystem of Linux desktop environments, Cinnamon occupies a unique space. Born from a desire for a familiar desktop experience amidst UI shifts, Cinnamon has carved its niche through a blend of traditional design, light-weight resource usage, and extensive customization options. This article delves into the history of Cinnamon, its core features, and how its implementations on these three distributions showcase its adaptability and appeal.

A Fork in the Road: From GNOME to Cinnamon

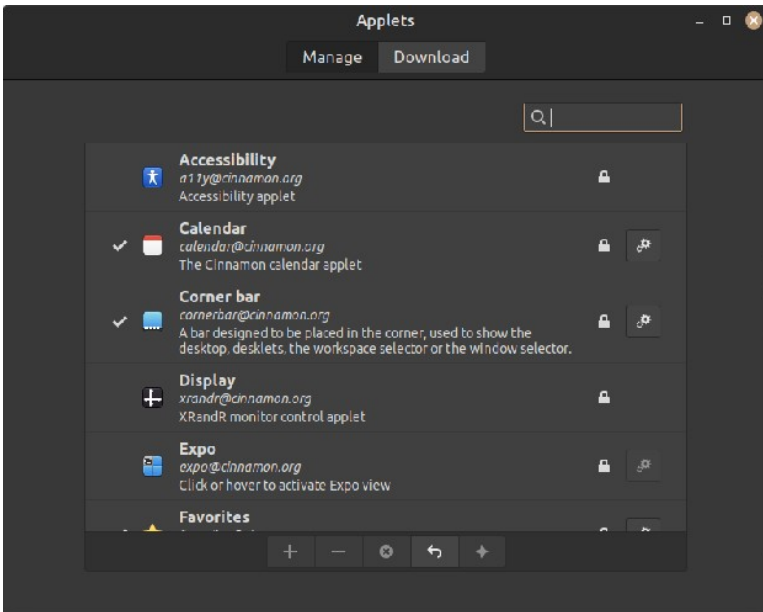
In 2011, the release of GNOME 3, with its radical interface shift, sparked discontent among users accustomed to the traditional desktop metaphor. Enter Linux Mint, a distribution renowned for user-friendliness, which chose to fork GNOME Shell and create its desktop environment, Cinnamon. The initial goal was to preserve the familiar panel-based layout, menus, and applets cherished by many Linux users.

Cinnamon quickly blossomed from a mere fork into a fully independent desktop environment. It retained the core GNOME technologies like GTK+ but built its components, including the Muffin [https://community.linuxmint.com/software/view/muffin] window manager, Nemo [https://en.wikipedia.org/wiki/Nemo_%28file_manager%29] file manager, and the Cinnamon Control Center, a fork of the GNOME Control Center.

This independence allowed Cinnamon to evolve at its own pace, prioritizing stability, user preferences, and resource efficiency.

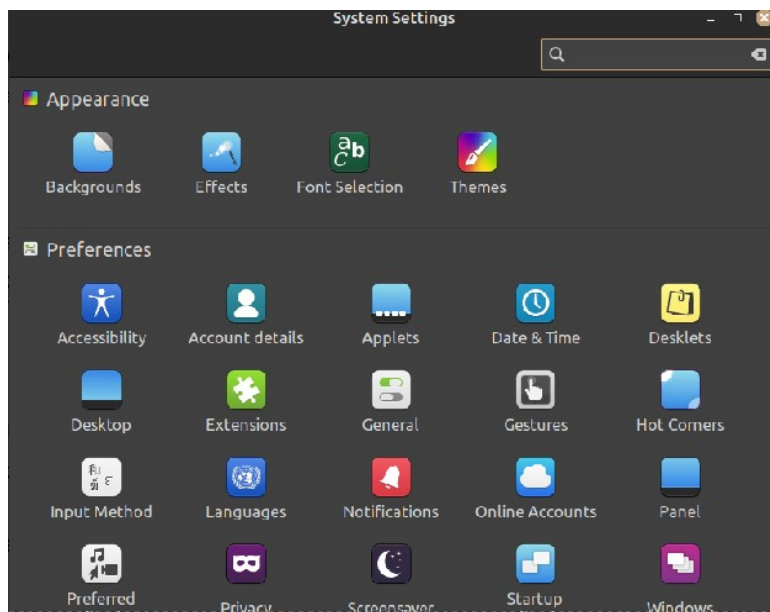
The Core of Cinnamon: Familiarity with a Flair

Cinnamon offers a desktop experience reminiscent of Windows XP and GNOME 2. At its heart lies a familiar panel encompassing a start menu, quick-launch icons, running applications, and system tray notifications. Users can customize the panel elements and their behavior and add applets for enhanced functionality. The applets include accessibility, display, corner bar, on-screen keyboard, panel launchers for favorite applications, workspace switcher, and many more.



Screenshot from Don Watkins' Linux Mint 21.2 system CC by SA 4.0

Beyond the traditional layout, Cinnamon boasts modern features like workspaces, desktop effects, and theme customization. The Cinnamon Control Center provides granular control over system settings, making it easy to personalize the experience. Resource efficiency remains a cornerstone, ensuring smooth operation on older hardware and a snappy user experience.



Screenshot from Don Watkins' Linux Mint 21.2 system CC by SA 4.0

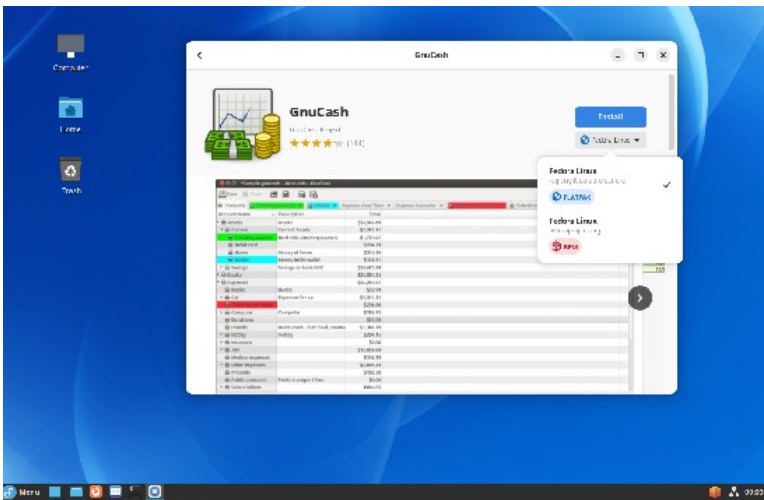
Cinnamon's Many Faces: Distributions and Distinctions

While adhering to its core principles, Cinnamon adapts to its surroundings across different distributions. Let's explore the nuances of Cinnamon implementations on Fedora, Ubuntu, and Linux Mint:

Fedora Cinnamon: Geared towards advanced users and developers, Fedora Cinnamon offers greater upstream integration and bleeding-edge software releases. Fedora Cinnamon allows

users to run on the latest kernel version, 6.5.6, in Fedora 39 with the BTRFS [<https://docs.kernel.org/filesystems/btrfs.html>] file system instead of EXT4 on Linux Mint and Ubuntu. Fedora uses `dnf` [https://github.com/manatools/dnfdragora] to load software by default. Do yourself a favor and install GNOME Software Center [https://en.wikipedia.org/wiki/GNOME_Software]. Once that is done, installing software on Fedora Cinnamon is much easier. Once the GNOME Software Center is installed, you can easily install 'rpm' or 'flatpak' software packages.

\$ `sudo dnf install gnome-software`.



*Screenshot from Fedora 39 Cinnamon desktop by Don Watkins
CC by SA 4.0*

Ubuntu Cinnamon: Ubuntu Cinnamon offers a good implementation of Cinnamon with support for the Snap Store [<https://snapcraft.io/store>] software center and traditional `.deb`

packages, either on the command line or from the Software Center. If you want to add software from Flathub, you'll have to install those packages [<https://flatpak.org/setup/Ubuntu>] yourself. Ubuntu Cinnamon supports a large number of desktop backgrounds and sets itself apart from Fedora there. Ubuntu Cinnamon runs on an EXT4 filesystem and Linux Kernel 6.2.0-39.

Linux Mint Cinnamon: As the birthplace of Cinnamon, Linux Mint offers the most polished and feature-rich experience. The default Mint themes and applets are meticulously designed, and the Cinnamon Control Center boasts even more granular customization options. Linux Mint Cinnamon 21.2 runs on Linux Kernel 5.15.0. Beyond familiarity, Linux Mint empowers users with Flatpak right out of the box. This flexibility, coupled with the comfort of a familiar desktop, creates a perfect recipe for Linux newbies to explore the boundless possibilities of open-source software.

Conclusion: A Spice of Choice for Every Palate

Cinnamon's adaptability and focus on user needs have won it a loyal following. Its implementations on Fedora, Ubuntu, Arch, Linux Mint Debian Edition, and Linux Mint exemplify its versatility, catering to different user choices within the Linux community. Whether you seek bleeding-edge software,

a familiar experience, or the most refined Cinnamon environment, each distribution offers a unique flavor of this popular desktop. Ultimately, Cinnamon's success lies in its commitment to familiarity, customization, and resource efficiency, offering a user-centric and visually appealing alternative in the diverse landscape of Linux desktops.

Getting Help for Linux

David Both

February 24, 2026

If you're a newcomer to Linux, or you are considering making the move to Linux, one question you'll most certainly ask is, "Is there help for Linux?" Other questions might be, "How good is Linux help?" and "How can I get help for Linux?"

The answers to those questions are, yes, excellent, and that's what this article is about.

I typically recommend Fedora and Linux Mint for new users and especially for anyone migrating to Linux from Windows, so I'll cover those two distributions in this article. Most distributions have excellent documentation, as do most of the open-source software available to run on Linux. But these are the two I know the best so they're the ones I'll cover.

The Fedora Project Docs [<https://docs.fedoraproject.org/en-US/docs/>] page is the starting point for all of Fedora's documentation. Linux Mint also has a documentation page

[<https://www.linuxmint.com/documentation.php>] that, while structured much more simply, also provides documentation that's more than sufficient for beginners and experienced users. Fedora's documentation is available in over 60 languages, and while Linux Mint supports over 100 languages, their web page for translations says that not all languages are fully supported yet. Both distros are recruiting translators to help.

I think the Linux Mint documentation is less complex and easier to navigate because it has fewer options than Fedora. Linux Mint has three editions to choose from, while Fedora has six editions, five atomic desktops, a "multitude" of spins and labs, as well as ARM and MAC images. The maze of links on the Fedora documentation page might be impenetrable unless you already know where you're going but I'll show you the secret doors.

Installation

Unless you have a Linux geek for a friend who'll do it for you, the first thing you're going to do is install Linux on your computer. You'll need to know how to download the Linux images from the Internet, create a bootable USB memory stick with them, and use that to install the distro you've chosen onto your computer.

Fedora

The Fedora Getting Started [<https://docs.fedoraproject.org/en-US/fedora/latest/getting-started/>] documentation that covers downloading and copying any of the several ISO images to a USB memory stick is quite good and offers at least a couple options for performing some of the steps involved. However, there's not a single word in any of the Fedora documentation I can find about how to actually perform an installation.

Fortunately the Fedora installation is easy, and Jim Hall, one of our editors and a frequent contributor to Both.org, has written articles about how to install the Fedora Xfce spin, which is what I usually recommend to people moving to Fedora. These instructions will also work for the other spins and most of the Fedora editions.

The Fedora 43 installation has been simplified to be as easy as the Linux Mint installation. It's now a matter of a few sequential steps that require some easy choices and a bit of data entry. Once that's done, the installation quickly proceeds through completion. Each of those steps is self-documenting on-screen, so there's really no need for additional documentation. Jim's article gives an excellent preview of what you'll need to do and the choices you can make. For most new Linux users it's all about taking the defaults and entering a passphrase if you choose to encrypt your storage drive.

Linux Mint

The Linux Mint Installation Guide is a complete guide to getting started, including choosing which of the three editions works best for you. The primary difference is the desktop: Cinnamon, Mate, or Xfce. You can read the installation guide to determine the differences.

The Guide also guides you through downloading the ISO image, creating a bootable Live USB drive with the ISO, and performing the installation. It also covers subjects like multi-booting with other operating systems, installing the notorious NVIDIA video drivers, multimedia codecs, creating system snapshots, disk partitioning, and more.

I suggest you read the entire document before performing the installation so you'll know what to expect before you start.

Like all of the Linux Mint documents, the Installation Guide is available in HTML for online viewing [<https://linuxmint-installation-guide.readthedocs.io/en/latest/>] and PDF [https://linuxmint-installation-guide.readthedocs.io/_/downloads/en/latest/pdf/] as well as ePUB for download.

Applications

All of the open-source applications you'll be likely to use have excellent documentation, either as part of the application or

online. I've used this documentation frequently and have usually found an answer to my questions.

You can usually press the F1 key to view the documentation or use the Help item on the app's menu bar. Some applications, like LibreOffice, even have printed documentation, which costs money but is very reasonable. The same documentation is also available in PDF for free downloading.

When the documentation doesn't have the information I need, I can go online to its website. Most Linux distributions and open-source applications have a website where you can ask questions or join a list where you can participate in discussions and ask questions.

Final thoughts

I find the Linux Mint documentation easy to navigate and well-done. It's clear and concise, with everything a new user needs to know. The installation guide does provide some advanced information but it's all aimed at helping new users get started.

The Fedora documentation is a little more difficult to navigate and spreads information around different pages. But it tends to be more technical and provides guidance for far more advanced options and potential problems than does the Linux Mint documentation.

Both of these distributions have lots of documentation available in one form or another. And the open-source applications that I use all have excellent documentation available at the push of a button, usually F1.

Linux and Security

David Both
May 23, 2025

Security is a critically important part of using and administering computers in our ultra-connected world. With Linux running almost all the Internet's infrastructure, including most websites, the world's financial foundation, and ever more desktop users, it has become a lucrative target for crackers.

Knowing and understanding the threats that jeopardize the safety and security of our computers is crucial for security, system, and network administrators.

There are two websites that collect information about cybersecurity vulnerabilities, and one of them focuses on Linux.

CVE: Common Vulnerabilities and Exposures

The Common Vulnerabilities and Exposures website [<https://www.cve.org/>] has been around for 25 years. The mission of the CVE® Program is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities. There are currently more than 280,000 CVE records in their database, which covers all operating systems and software.

CVE partners and end users can report vulnerabilities, which are given identifiers, such as CVE-2025-1234, and entered into the CVE database. The database is open and free to search by anyone. Anyone can also submit reports of vulnerabilities for possible inclusion in the database.

The CVE website provides educational and informative links that cover specific vulnerabilities as well as more general security best practices. It also provides downloads of the complete CVE database for local use.

Some general news organizations and several cybersecurity-focused newsletters and websites report on some of the most dangerous vulnerabilities, but checking the site for the most recent entries is important to stay fully informed. They also have a newsletter

CVE is one of the most critical pillars of cybersecurity infrastructure on the planet and it's currently in trouble. One of the CVE sponsors is the U.S. Department of Homeland Security (DHS) Cybersecurity and Infrastructure Security Agency

(CISA) [<https://www.dhs.gov/>], which, due to the idiots in the U.S. government, cut off funding of CVE for what turned out to be a short time. This time, funding was restored after a few days, but CVE itself is vulnerable in these tumultuous times.

Linux Security

The Linux Security [<https://linuxsecurity.com/>] website focuses on vulnerabilities that affect Linux and open-source software. It contains news, advisories, and how-to articles that can provide guidance in enhancing the security of Linux hosts and networks. The Linux Security organization has been around since 1996.

Their news, advisories, and must-read articles highlight the most critical threats and security issues. They also have email newsletters and I subscribe to them to ensure that I am getting the very latest information. I've used the information in the newsletter to verify that my systems are as secure as possible on several occasions. I typically get two to four newsletters a week. They seem to send them as necessary rather than on a set schedule. I like the newsletters as an alert system because they describe the threats in detail along with mitigation procedures.

Parting Thoughts

Security for Linux systems is becoming more critical as the threat profiles grow. Having knowledge about the specific threats and the methods to mitigate them is an important part of our cybersecurity toolkit. These two websites provide information and insights that allow us to harden our Linux systems as much as possible.

Linux Desktops and Laptops

Jim Hall

May 20, 2024

I've run Linux on the desktop since 1993. In that time, I've lost track of how many machines I've run Linux on, either at home or in the office. But I'll always remember the "milestone" machines. My first Linux computer ran on a '386-SX desktop. Much later, I bought a VA Linux mini-tower. At work, I always ran Linux on a laptop. These days, I run Linux on several machines, including a laptop, desktop, and Raspberry Pi.

I wanted to know how others ran Linux, either at home or at work. I asked our community to find out the following:

Chris Hermansen runs several machines:

"I run the latest Ubuntu, whether LTS (Long Term Support) or intermediate versions. Right now that's Ubuntu 24.04, which is an LTS version. I run this on three systems:

My work laptop, which is a System76 Gazelle from 2019 with an Intel i7 8th-generation CPU.

My work desktop, which is a homebuilt Gigabyte Technology X570 Aorus Pro Wifi motherboard with an AMD Ryzen7 3800X CPU and an NVIDIA GeForce RTX3060 graphics card.

And a spare old desktop from approximately 2007, which has a Gigabyte motherboard of some sort and an Intel Core Duo CPU. (This runs just fine, by the way. We keep our secondary copy of our digital photos there.)

I have used Ubuntu as my work and non-work operating system since version 4.10 in 2005. Back then it was running on a lovely-at-the-time Toshiba laptop, which was my first non-Sun computer since 1986.”

Alan Formy-Duval built his own machine:

“My desktop system is one that I built myself. I have always built my own since my days as a Value Added Reseller working in PC shops in the 1990s. Currently it is running Fedora Linux Workstation Edition. Hardware specs are:

Microstar (MSI) Mortar Z270M motherboard, Intel Core i7-7700K CPU, 24 GB RAM, AMD Radeon RX 6650 XT graphics, and Samsung SSD 970 EVO Plus disk

It is about eight years old, but it runs great given Linux goes easy on system resources.”

David Both also builds his own Linux computers:

“I usually build my own desktop computers. Out of twelve total computers, nine are desktops, and I have built eight of those. One is a very old Dell that was gifted to me and that I

am using for longevity testing. I also have three laptops, all of which I purchased from System 76.

My primary workstation is built on a Cooler Master MasterFrame 700 frame. It has an ASUS TUF X299 MARK 2 motherboard and 64 GB of RAM because I run multiple virtual machines simultaneously for testing with an Intel Core i9-7960X CPU @ 2.80 GHz with 16 cores and HT for 32 CPUs, that I've overclocked to 3.8 GHz, and is liquid-cooled. It also has an AMD Radeon PRO WX2100 GPU. I also use a Corsair K70 RGB MK2 keyboard and an Ergo M575 wireless mouse.

My primary laptop is a System76 Oryx Pro8 with an Intel Core i7-10875H CPU @ 2.30 GHz CPU with 16 cores and 32 CPUs overclocked to 5.1 GHz and 64 GB of RAM; again, so I can run multiple virtual machines.

The old Dell is an Optiplex GX620 that was built in 2005. It has an Intel Pentium 4 CPU at 3.00 GHz with a single core, 2 CPUs, and 4 GB of RAM.

I use another of the less powerful ones as my server. This hosts my email server, three websites, and a number of lists. This computer uses an Intel Core i7-8700 CPU @ 3.20 GHz that has 6 cores and 12 CPUs, with 32 GB of RAM.

All my computers run "24x7" on Fedora 40."

Kevin Sonney uses Linux all around the house:

"I don't use Linux as my "primary" desktop, but I do use it on a lot of machines here in my house (mostly Raspberry Pis). However, I do have an old Lenovo all-in-one running Fedora Linux as the control panel for a laser engraver and a

Thinkpad X250 as a multi-boot “test” box for Fedora Linux and ArcaOS (ArcaOS isn’t Free Software, I know, but I’m one of those weird OS/2 people).”

Don Watkins likes the NUC for a micro-desktop:

“My primary workspace revolves around my trusty Intel NUC 11, a powerhouse packed into a compact form factor. This miniature marvel boasts a quad-core 11th Gen Intel Core i7 processor coupled with a whopping 64 GB of memory. Rendering visuals is a breeze thanks to the Intel TigerLake-LP GT2 Iris Xe Graphics it’s equipped with. For video conferencing, I rely on the AVerMedia Live Streamer CAM 313, seamlessly integrated into my setup.

What I adore most about the NUC is its minimal footprint, fitting snugly into my workspace without overwhelming it. Connectivity is top-notch with Intel WiFi 6 and Intel Gigabit Ethernet, ensuring I’m always connected. Storage needs are met by the Samsung 980 Pro NVME drive, offering a generous terabyte of space. Operating on the reliable Linux Mint Virginia 21.3, my setup is both powerful and efficient.

Accompanying my NUC desktop is the faithful DarterPro from System76, a workhorse that has served me well over the past five years. Sporting an Intel i7 processor and 16 GB of RAM, it’s a reliable companion for when I’m on the move. Like my NUC, it also runs Linux Mint, ensuring consistency across my devices.

In my journey through various Linux distributions, Linux Mint has emerged as my steadfast choice for the past four years. The allure of the Cinnamon desktop environment

has captivated me, its intuitive design seamlessly blending with my workflow. Linux Mint's native support for flatpak eliminates the need for additional tweaks, streamlining my experience further.

Aiding others in transitioning to Linux is a passion of mine, and Linux Mint's familiar interface simplifies this process. With Firefox and Google Chrome as my trusted browsers, I navigate the digital landscape effortlessly. GnuCash remains a cornerstone of my financial management, serving both personal and professional needs, including my role as a fraternity treasurer.

I rely on the dynamic duo of Jitsi and Zoom for video conferencing, both impeccably supported by Linux Mint. Joplin caters to my note-taking requirements, while Visual Studio Code and LibreOffice handle word processing and light database work seamlessly.

My Linux ecosystem, centered around the NUC and complemented by the DarterPro, embodies efficiency, reliability, and familiarity. With Linux Mint as my guiding light, I navigate the digital realm with ease, equipped with tools tailored to enhance productivity and streamline workflows."

Jim Hall uses a laptop and a desktop:

"For myself, I use a Lenovo ThinkCentre M720 Tiny desktop that I bought in 2019, running Fedora Linux. This has an 8th-generation Intel Core i3-8100T CPU at 3.10 GHz, 32 GB of memory, and a 256 GB PCIe-NVME Solid State Drive for storage. I used to run Fedora with GNOME, but since Fedora 40, I'm running Xfce as my desktop.

When I'm on the go, I run a 12-year-old (2012) "1st Gen" Lenovo X1 Carbon laptop with an Intel Core i5-3317U CPU. It has 4 GB of memory (which is small these days) and 128 GB Solid State Drive for storage. This is also running Fedora Linux, with Xfce.

I also have a Raspberry Pi 3B that I use as a kind of "mini-server" to share files, access my printer, and use as a "development" hub to build websites. It runs Fedora Server."

My First Linux System

Jim Hall

August 25, 2025

When I went to university in the early 1990s, I discovered our Unix computer lab. I immediately fell in love with the Unix command line, which was both somewhat *familiar* (due to my DOS experience) and *powerful* (because of all the commands that did things that DOS never imagined, especially with processing and manipulating text files).

It was with that context that I was so excited in 1993 to learn about this new thing called *Linux*, a free version of Unix that I could run on my '386 computer at home. I immediately wanted to try it out.

Softlanding Linux System (SLS)

My first Linux distribution was Softlanding Linux System (SLS) 1.03, with Linux kernel 0.99 alpha patch level 11. My '386 had just enough horsepower that I could install and run Linux.

A year later, I upgraded to SLS 1.05, with the brand-new Linux kernel version 1.0, which introduced this new idea called “*kernel modules*.” With modules, you no longer needed to completely recompile your kernel to support new hardware; instead, you just loaded a kernel module (there were 63 of them at the time). SLS 1.05 included this note about modules in the distribution’s README file:

Modularization of the kernel is aimed squarely at reducing, and eventually eliminating, the requirements for recompiling the kernel, either for changing/modifying device drivers, or for dynamic access to infrequently required drivers. More importantly, perhaps, the efforts of individual working groups need no longer affect the development of the kernel proper. In fact, a binary release of the official kernel should now be possible.

On August 25, Linux turns 34 years old. To celebrate, I reinstalled SLS 1.05 to remind myself what it was like to run Linux for the first time, all those years ago.

Installing SLS

SLS was the first Linux distribution that you could call a *distribution*. Before then, if you wanted to install Linux on a new PC, you basically had to make an image of a working Linux system and copy that to the new computer. SLS provided an install program. To be fair, it was a very basic installer, but it did the job. To start, you boot the install floppy and then login as the install user.

```
NE*000 ethercard probe at 0x300: 52 54 00 12 34 57
eth0: NE2000 found at 0x300, using IRQ 9.
me.c:v0.99-15k 3/3/94 Donald Becker (becker@super.org)
Checking 386/387 coupling... Ok, fpu using exception 16 error reporting.
Linux version 1.0 (root@softland) #1 Tue Apr 26 19:25:39 PDT 1994
Partition check:
  hda: hda1 hda2
RAMDISK: 1474560 bytes, starting at 0x1e950c
RAMDISK: Loading 1440 blocks into RAM disk.....
done
MINIX-fs: mounting unchecked file system, running fsck is recommended.
UFS: Mounted root (minix filesystem).
Jul 22 21:53:14 init[1]: version 2.4 booting
/proc on /proc type proc (defaults)
Jul 22 21:53:14 init[1]: Entering runlevel: 4

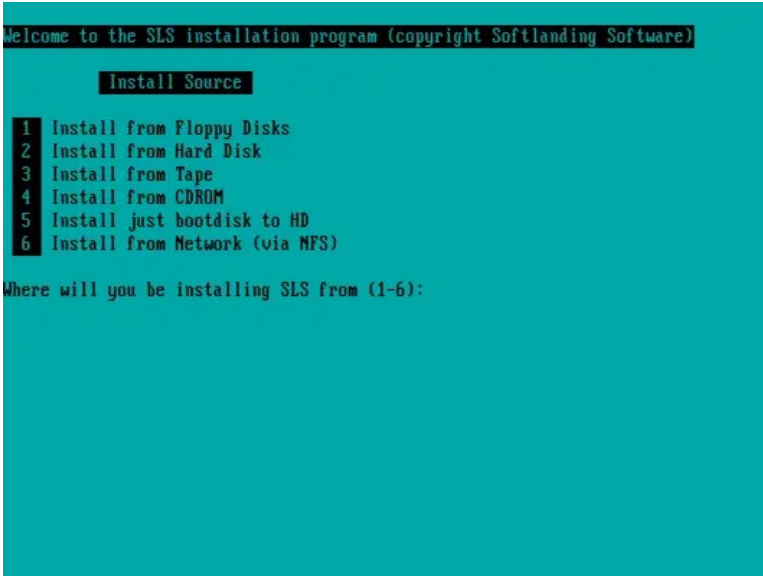
Welcome to the SLS (copyright Softlanding Software) install/boot disk.
Redistribution is hereby granted, providing compliance with GPL is observed,
that Softlanding is acknowledged (as above) in advertisements, and
that the terms of support (or lack thereof) are explicitly stated.

Login now as "install" to begin installation. Or login as "root" and
enter "install.info" for information on installation and partitioning.
softland login:
```

Starting the installer from floppy disk

A neat feature introduced in SLS 1.05 was the color-enabled text-mode installer. When I selected color mode, the installer switched to a light blue background with black text. This didn't really offer much of an experience, but I guess it was

better than the plain white-on-black text used in previous versions.

A screenshot of a terminal window showing the SLS installation program. The background is a solid teal color. The text is white. At the top, it says "Welcome to the SLS installation program (copyright Softlanding Software)". Below that, a header "Install Source" is displayed. A numbered list of six options is shown: 1 Install from Floppy Disks, 2 Install from Hard Disk, 3 Install from Tape, 4 Install from CDROM, 5 Install just bootdisk to HD, and 6 Install from Network (via NFS). At the bottom, it asks "Where will you be installing SLS from (1-6):".

```
Welcome to the SLS installation program (copyright Softlanding Software)

  Install Source

  1 Install from Floppy Disks
  2 Install from Hard Disk
  3 Install from Tape
  4 Install from CDROM
  5 Install just bootdisk to HD
  6 Install from Network (via NFS)

Where will you be installing SLS from (1-6):
```

Starting the installer in color mode

```
Now put a formatted floppy into your boot drive and hit enter:
645+1 records in
645+1 records out
Do you wish to have the video mode preset at boot time? [n]: n
do you have a modem (y/n): n
Setting up lilo on /dev/hda2
Is it ok to make /dev/hda2 the active partition? [y]: y
If any, specify a DOS partition to setup for booting (eg: /dev/hda1): /dev/hda1
Added linux
Added linux2
Added dos
harddrive is now setup for booting
Host name to give to this machine [softland]: sls105
Enter your ip number: 127.0.0.1\1.0.0.721/192.168.1.100
Ip address of network [192.168.1.0]:
Ip address of router [192.168.1.1]:
Domain name that this machine [softlanding.com]: example.com
Do you wish to have your machine setup as a name server? (y/n): n
Enter ip number of nameserver host: 8.8.8.8/8.8
Domain name of nameserver host [example.com]:
please stand by: setting up system file permissions
Finished setting up system file permissions
If there is a patch disk to install, insert it in A: and type y [n]: n
Installation is complete. Hit enter to reboot your computer now...
```

Finishing the installation

SLS was released in the 1990s, when distributions were provided on floppy disk. That's how I installed SLS this time, from a stack of virtual floppy images. This was a bit of a pain to install in a virtual machine, because I had to swap out each of the floppy images in the virtual machine every time the installer asked for a different floppy, but eventually I got there.

By responding to a few simple prompts, I was able to create a partition for Linux, put a filesystem on it, and install SLS. When I finished the install and booted into the SLS login prompt for the first time in so many years, I felt transported back in time.

```
Floppy drive(s): fd0 is 1.44M
Floppy: FDC version 0x90
Swansea University Computer Society Net2Debugged [1.30]
IP Protocols: ICMP, UDP, TCP
NE*000 ethercard probe at 0x300: 52 54 00 12 34 57
eth0: NE2000 found at 0x300, using IRQ 9.
ne.c:v0.99-15k 3/3/94 Donald Becker (becker@super.org)
Checking 386/387 coupling... Ok, fpu using exception 16 error reporting.
Linux version 1.0 (root@softland) #1 Tue Apr 26 19:25:39 PDT 1994
Partition check:
  hda: hda1 hda2
EXT2-fs warning: mounting unchecked fs, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem).
Cannot open *.o
[!: argument expected
none on /proc type proc (rw)
/etc/rc.net: sls105(192.168.1.100), rpc.portmap, inetd, namedJul 22 13:57:32 nam
ed[45]: restarted

, rpc.nfsd, rpc.mountd

Welcome to Linux 0.99.15g
sls105 login:
```

Booting SLS 1.05 for the first time

Trying it Out

Let's highlight a few things about this early Linux distribution. Linux 1.0 didn't need a lot of overhead, requiring just 2 MB of RAM to boot, or 4 MB if you want to compile programs, or 8 MB to run the X Window System. And the distribution didn't take much room, either. SLS 1.05, including the X Window System and development tools, used about 85 MB of disk space. That may not sound like much space by today's standards, but when Linux 1.0 came out, 120 MB hard drives were still common.

```
jimhall$ uname -a
Linux s1s105 1.0 #1 Tue Apr 26 19:25:39 PDT 1994 i686
jimhall$ free
              total         used         free      shared    buffers
Mem:          14972         3544        11428         2004        1260
Swap:           0             0             0
jimhall$ df -h
df: illegal option -- h
Usage: df [OPTION] [PATH]...

Try 'df --help' for more information.
jimhall$ df -k
Filesystem      1024-blocks  Used Available Capacity Mounted on
/dev/hda2         112896    86762    25006      78% /
jimhall$ _
```

df shows only about 85 MB disk space used

The familiar /proc meta filesystem exists in Linux 1.0, although it doesn't provide much information compared to what you see in modern systems. In Linux 1.0, /proc includes interfaces to probe basic system statistics like *meminfo* and *stat*.

```
jimhall$ cd /proc/
jimhall$ ls
1/      44/    65/    71/    75/    meminfo  stat
107/   48/    67/    72/    kcore  modules  uptime
15/    52/    69/    73/    kmsg   net/     version
40/    56/    70/    74/    loadavg self/
jimhall$ cat stat
cpu 170 0 166 16649
disk 0 0 0 0
page 6389 179
swap 0 0
intr 30693
ctxt 12795
btime 2640071119
jimhall$ cat version
Linux version 1.0 (root@softland) #1 Tue Apr 26 19:25:39 PDT 1994
jimhall$ cat meminfo
          total:    used:    free:    shared:    buffers:
Mem: 15331328  9162752  6168576  2023424  6569984
Swap:          0          0          0
jimhall$
```

Using the /proc filesystem

The /etc directory on this system is pretty bare. Notably, SLS 1.05 borrowed the *rc* concept, where the system loads everything from *rc* scripts, with local system changes defined in the *rc.local* file. It was a simpler time when you could easily modify what got loaded just by editing a few files.

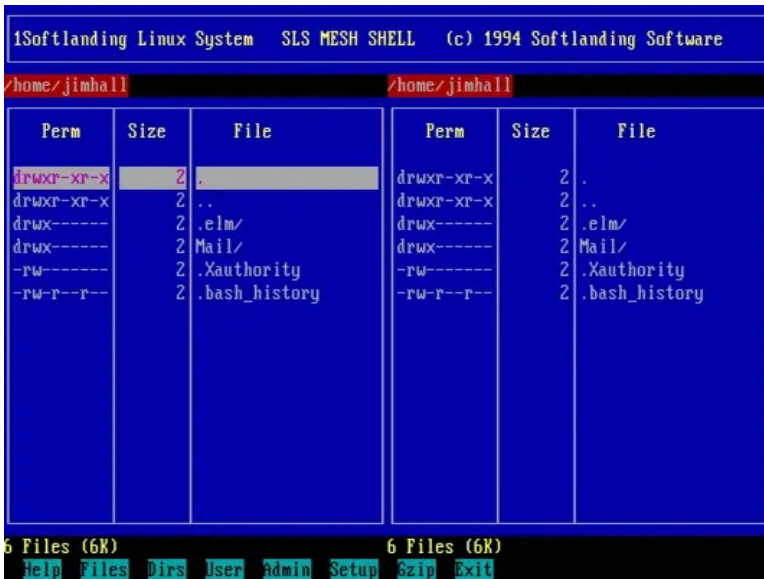
```
jimhall$ cd /etc
jimhall$ ls
DIR_COLORS      group.bak      lilo.conf      netlist        securetty
HOSTNAME        gshadow        lilo.conf.in  netlist.in    services
NETWORKING     host           login.defs     networks      shadow
bootptab       host.conf      magic          nextserial*   shadow.OLD
csh.cshrc      hostcvt.build* meshrc         nntpserver    shells
csh.login      hosts          minicom.users passwd         skel/
default/       hosts.allow   minirc.dfl    passwd.OLD    soabasefile
diphosts       hosts.bak     motd          passwd.bak    soabasefile.in
disktab        hosts.deny    mount*        printcap      syslog.conf
domain         hosts.equiv   mtab          profile*      syslog.pid
exports        hosts.in      mtools        profile.sw    termcap
fdprm          hosts.lpd     named.boot    protocols     ttys
fs/            huconfig     named.boot.in rc*           ttytype
fstab          inet@        named.hosts   rc.local      umount*
ftpaccess      inetd.conf   named.local   rc.mod        utmp
ftputers      inittab      named.pid     rc.net*       wtmp@
gateways       issue        named.reload@ resolv.conf
gettytab       ksh.kshrc   named.restart@ rpc
group          lilo.bootsave named.rev      sample.dip
jimhall$
```

There's not much in /etc

Back to Work

With my system up and running, it was time to see what I could do with it. In the 1990s, SLS was pretty cutting-edge. While my DOS system had many more “mature” applications like Lotus 1-2-3 and As-Easy-As for spreadsheets, WordPerfect and Galaxy Write for word processing, ProComm and Telix for dial-up access, and a host of other applications and games, Linux didn't have a lot to offer at the time. But it had enough to show *promise*.

Let's start with basic file management. SLS included the Soft-landing Menu Shell (MESH), a file management program similar to Midnight Commander. Users in the 1990s might have compared MESH more closely to Norton Commander, a popular file manager available on MS-DOS.



MESH file manager

Aside from MESH, SLS included a few other full-screen applications, including ELM to read email, GNU Emacs, and the venerable Vim editor.

Introduction to Linux

```
GNU Emacs 19.19.2 of Sun Sep 5 1993 on softland (linux)
Copyright (C) 1993 Free Software Foundation, Inc.

Type C-h for help; C-x u to undo changes. ('C-' means use CTRL key.)
To kill the Emacs job, type C-x C-c.
Type C-h t for a tutorial on using Emacs.
Type C-h i to enter Info, which you can use to read GNU documentation.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
You may give out copies of Emacs; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version._

-----Emacs: *scratch* (Lisp Interaction)--All-----
```

GNU Emacs 19

```
Mailbox is '/var/spool/mail/jimhall' with 0 messages [ELM 2.4 PL21]

You can use any of the following commands by pressing the first character;
d)delete or u)ndelete mail, m)ail a message, r)eply or f)orward mail, q)uit
To read a message, press <return>. j = move down, k = move up, ? = help

Command:
```

ELM 2.4 for email

SLS also offered a few games, including a terminal-based version of Tetris. If you were alive in the 1990s, you didn't have to go far to see a version of Tetris running *somewhere*.

```

||                                     || TETRIS FOR TERMINALS
||                                     ||
||                                     || Written by Mike Taylor
||                                     || Email: mirk@uk.ac.warwick.cs
||                                     || Started: Fri May 26 12:26:05 BST 1989
||                                     ||
||                                     || Game level: 0
||                                     ||
||                                     || Score: 38
||                                     || Pieces: 8
||                                     || Levels: 1
||                                     ||
||                                     || Use keys:
||                                     || =====
||          [][]  || Move left:  'j'
||          [][]  || Move right: 'l'
|| <>             || Rotate:    'k'
|| <>  /xxx@     || Drop:      ' '
|| <> /xxx@e@e{}{} || Pause:    's'
|| <> /xxx@e{}{}## || Quit:     'q'
|| +=====+     || Refresh:  '^L'

```

Tetris for terminals

In the 1990s, home users were more likely to talk to the outside world over a modem. SLS 1.05 included the Minicom terminal application, which gave a direct connection to the modem, such as the AT commands to dial a number or hang up the line. Minicom also supported macros and other neat features to make it easier to connect to your local modem pool. As an undergraduate student in the 1990s, I used Minicom *all the time* to dial into the campus computer network.

A screenshot of a terminal window with a black background and white text. The text at the top reads "Minicom 1.5 Beta Copyright (c) 1993 Miquel van Smoorenburg" followed by "Press CTRL-A Z for help on special keys". A horizontal line is visible below the text. At the bottom of the terminal, a status bar displays "CTRL-A Z for help | 38400 8N1 | NOR | Minicom 1.5B 1993 | VT100 | OFFLINE".

```
Minicom 1.5 Beta Copyright (c) 1993 Miquel van Smoorenburg
Press CTRL-A Z for help on special keys

-

CTRL-A Z for help | 38400 8N1 | NOR | Minicom 1.5B 1993 | VT100 | OFFLINE
```

Minicom 1.5 beta for dial-up

Unix systems didn't have desktop word processors at the time. It would be years before StarOffice (the proprietary predecessor to OpenOffice, later LibreOffice) would be available for Linux systems. Instead, if you wanted to write a document, you used a *document preparation system* like LaTeX, nroff, or troff.

Actually, troff was for *phototypesetters*, but GNU provided the groff workalike since the late 1980s, which produced Postscript output. At the time, the typical way to print nice-looking groff output was to use GNU Ghostscript to convert and print it on my Epson dot matrix printer. But this was very slow and required a lot of printer ribbon to cover the several passes required for high-resolution output on my 9-pin printer. Most of

the time, I used the nroff processor instead, which used plain text that I could print anywhere very quickly.

As an undergraduate physics student, I also used LaTeX to write lab reports that required formatting equations; I would print LaTeX documents on my printer in the same way. With Ghostscript, though, I would sometimes spend a few bucks to print it on the laser printer in the campus computer lab instead.

```
.pp
When I was just getting started in Unix and Linux in the early 1990s,
we didn't have fancy word processors like we do today. On DOS systems,
we had WordPerfect and Galaxy Write, which did a very good job. But
these didn't exist on Unix.
.pp
So to write my class papers when I was a college student, I could
either boot to DOS and use a word processor there, or I could teach
myself how to use a text markup system like roff. We also had LaTeX,
but roff was quick and easy to type up a simple paper like for English
class. I usually reserved LaTeX for when I needed scientific markup,
like writing lab reports for physics classes.
.pp
With nroff, you could easily write a paper with simple markup, then process it \
using
.b nroff
.b -me
.b paper.me
and it would magically turn into something you could send to the
printer. nroff could even do basic formatting like
.b bold
and
.i italics
-----Emacs: paper.me (Nroff)--Top-----
Loading nroff-mode...done
```

Writing a paper in nroff -me

```
When I was just getting started in Unix and Linux in
the early 1990s, we didn't have fancy word processors like
we do today. On DOS systems, we had WordPerfect and Galaxy
Write, which did a very good job. But these didn't exist on
Unix.

So to write my class papers when I was a college stu-
dent, I could either boot to DOS and use a word processor
there, or I could teach myself how to use a text markup sys-
tem like roff. We also had LaTeX, but roff was quick and
easy to type up a simple paper like for English class. I
usually reserved LaTeX for when I needed scientific markup,
like writing lab reports for physics classes.

With nroff, you could easily write a paper with simple
markup, then process it using nroff -me paper.me and it
would magically turn into something you could send to the
printer. nroff could even do basic formatting like bold and
italics text.

line 5
```

Viewing the formatted output

Running the X Window System

Configuring the X Window System by hand really can be a sobering experience. The X Window System could support a variety of video cards and monitors, and setting up the X Window System for the first time required a lot of experimenting with sync rates and resolutions until you settled on something that worked. SLS 1.05 made this a little less painful by providing a “helper” program to identify and define various system settings for you. After a few prompts and some tweaking, I was finally able to launch the X Window System.

```
Softlanding Software (604) 592-0188, gentle touch downs from DOS bailouts.
Welcome to Linux SLS 1.05. Type "mesh" for a menu driven interface.
Fresh installations should use "syssetup" to link the X servers, etc.

1:sls105:/$ syssetup

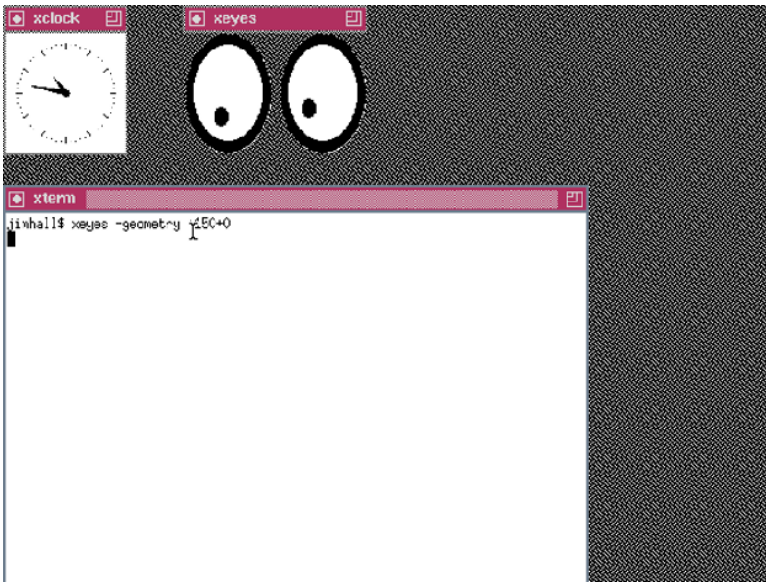
  0 - quit
  1 - Mouse
  2 - Modem
  3 - Lilo
  4 - Hostname
  5 - X-Windows

Which item to setup? [1-5] 5
0 - Skip setup
1 - Logitech BusMouse
2 - PS2 Bus Mouse
3 - Microsoft Bus Mouse
4 - ATI XL Bus Mouse
5 - Microsoft Serial Mouse
6 - Logitech Serial Mouse
7 - MouseSystems Serial Mouse
8 - MM Serial Mouse

What type of mouse do you have (0-8):
```

Setting up the X Window System

But this is from 1994, and the concept of a *desktop* didn't exist yet; KDE wouldn't arrive until 1996, and GNOME until 1999. In 1994, I used TWM until I was able to upgrade my home system, and then I switched to FVWM. TWM is a simple, no-frills, functional, graphical environment, but it was also much easier to set up, so that's what I used here.



TWM with default colors

A Blast From the Past

As much as I enjoyed exploring my Linux roots, eventually it was time to return to my modern desktop. I originally ran Linux on a 32-bit '386 computer with just 8 MB of memory and a 120 MB hard drive. That wasn't bad for the era, but it's very pokey compared to today. I should note that SLS runs *much faster* in a modern virtual machine, especially one that's hosted on my dual-core, 64-bit Intel Core i5 CPU with 4 GB of memory and a 128 GB solid-state drive. You don't really get to experience the delay of typing a big command before you see the

output or starting a build with make and watching the compiler slowly grind through all of the source files.

But running this original version of SLS served as a great reminder of how far we've come with Linux. Back in the 1990s, Linux was a great step forward for folks who ran big Unix systems—but not much of a step up from more mainstream “desktop” operating systems like DOS or Mac. Today, Linux is more than a Unix replacement; it's a modern desktop environment that's easy to use and filled with apps. So while I enjoyed this look back at Linux from 1994, I was also glad to return to Linux in 2025.

```
l:sls105:/$ ls
Image      home/      linux@    proc/     usr/
bin/       install/   local@    root/     usrlocal/
boot/      interviews/ lost+found/ sbin/    var/
dev/       lastlog@   ls@       tmp/      zImage
etc/       lib/       mnt/      user/     zSystem.map
l:sls105:/$ shutdown -h now
Why? time to leave

URGENT: broadcast message from root:
System going down IMMEDIATELY!

... time to leave ...

Now you can turn off the power...
```

Shutting down the system

System76 and the New COSMIC Desktop

Don Watkins

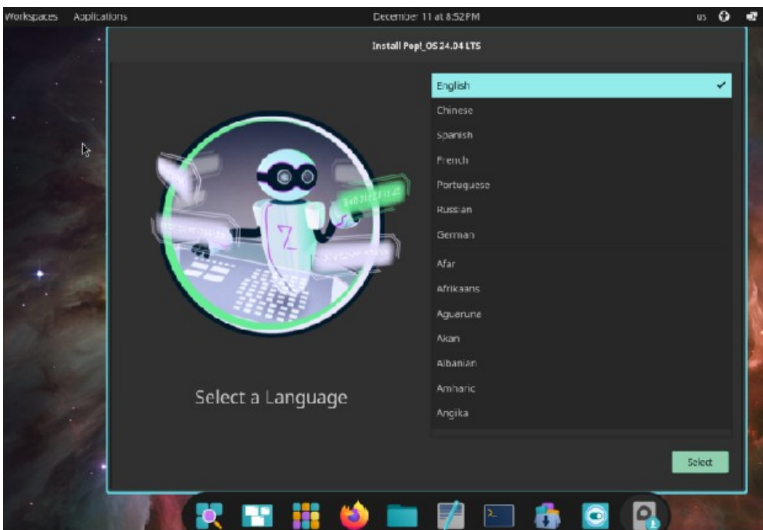
December 12, 2025

System76 is celebrating 20 years, and they did so by officially releasing Pop!_OS 24.04 LTS [<https://system76.com/pop/>] and the COSMIC desktop. I got an email from Carl Richell today that said, “COSMIC is built on the ethos that the best open-source projects enable people to not only use them, but to build with them. COSMIC is modular and composable. It’s the flagship experience for Pop!_OS in its own way and can be adapted by anyone who wants to build their own unique user experience for Linux.”

I have been following the development of COSMIC for the past four years. System76 had been customizing GNOME for Pop!_OS, which first launched on October 27, 2017, with Ubuntu 17.10. The operating system was designed to provide a clean, productivity-focused experience, featuring full-disk encryption by default, curated drivers for NVIDIA GPUs, and a streamlined installer. Over time, System76 introduced its

own desktop environment tweaks, which ultimately led to the development of the COSMIC desktop. This project began in 2021 and has become fully independent.

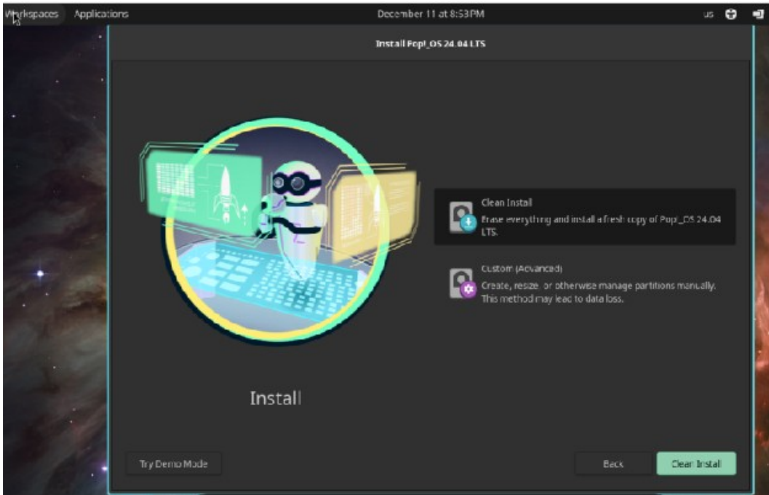
Today, I received an email announcing the release of COSMIC as Pop!_OS. I was excited to download it and try it out on a spare laptop that has an i7 processor and 16 GB of RAM. I downloaded the ISO file, created a bootable drive, and started the installation process. As usual, it was extremely easy and well-designed. The installation proceeded just like previous Pop!_OS installations.



Screenshot by Don Watkins CC-by-SA 4.0

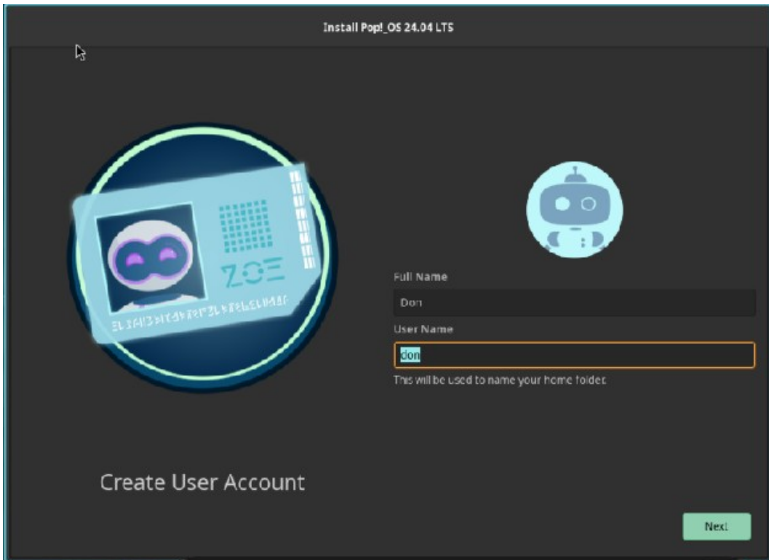
I decided to do a clean install, erasing the recently installed Fedora KDE Plasma.

Introduction to Linux



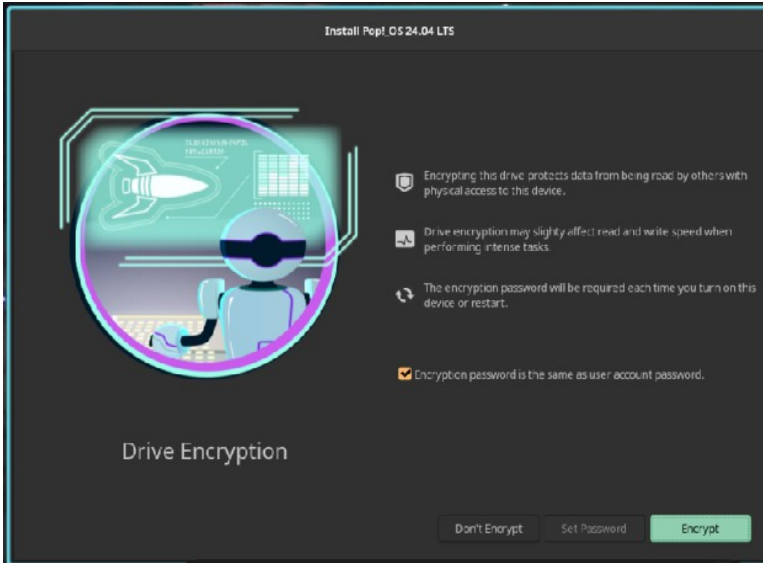
Screenshot by Don Watkins CC-by-SA 4.0

I was invited to create a user and provide a password.



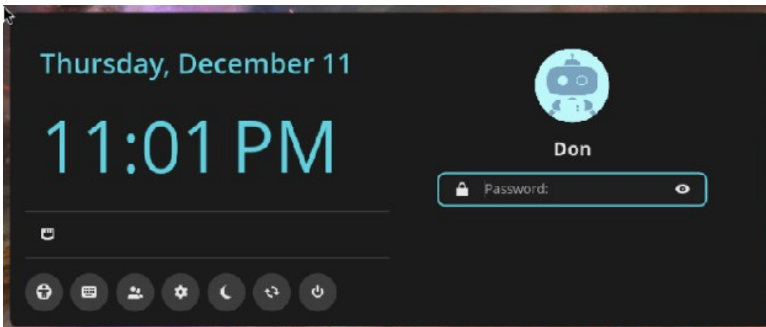
Screenshot by Don Watkins CC-by-SA 4.0

I chose not to encrypt my disk, but it's configured to use your user password as the encryption password unless otherwise specified.



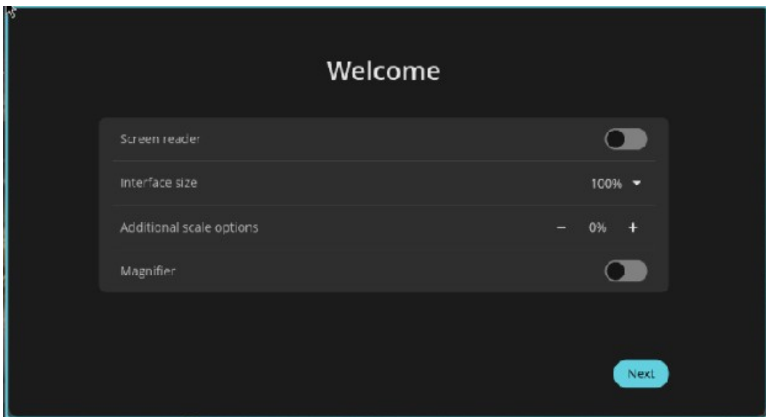
Screenshot by Don Watkins CC-by-SA 4.0

Installation proceeded quickly on my laptop. When the initial install was finished, I was invited to restart the computer. That's when I began to see the changes brought by the COS-MIC desktop.



Screenshot by Don Watkins CC-by-SA 4.0

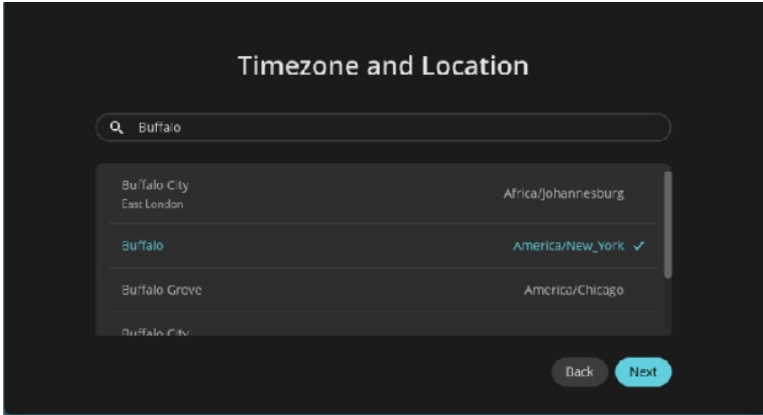
Wow, this was a new feature! At the bottom-left of the initial login screen, I noticed I could set my accessibility options, including a screen reader, magnifier, high contrast, and inverted colors. This is such an important option for those of us who need accessibility options from the first login.



Screenshot by Don Watkins CC-by-SA 4.0

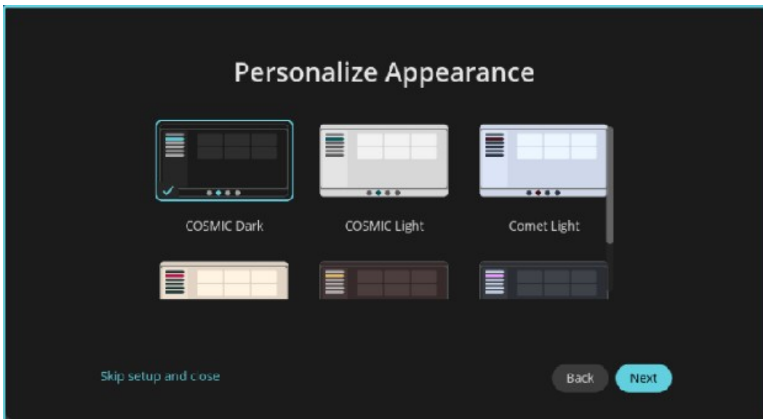
On the next screen, I can set the screen reader, interface size, additional scale options, and the magnifier. Kudos to System76 for putting accessibility first in COSMIC. The next

screen is my chance to connect to the wireless networks I will be using. Following that is the choice of language I will be using. Then, a screen to set the keyboard layout, time zone, and location.



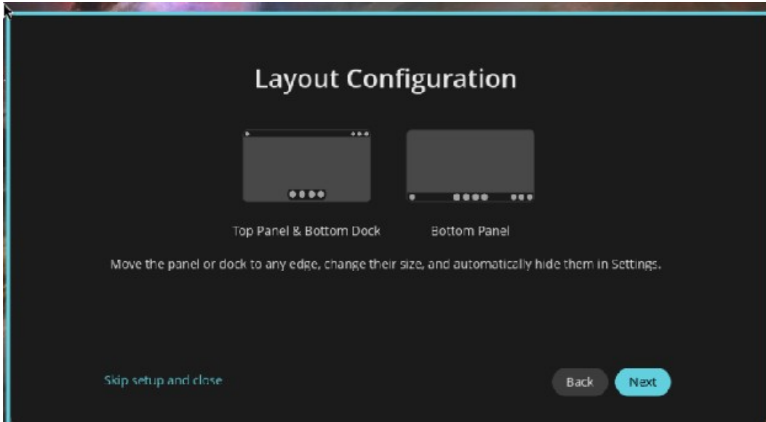
Screenshot by Don Watkins CC-by-SA 4.0

Next, I am provided with options for how I want my desktop to look. This is too cool.



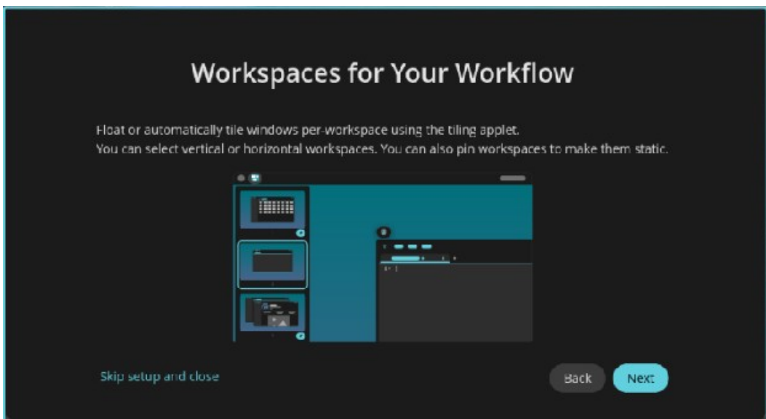
Screenshot by Don Watkins CC-by-SA 4.0

Next, I can choose how I want the dock to appear on my desktop. They really considered all the options I might want to customize my workspace, giving me maximum flexibility to tailor my workflow.



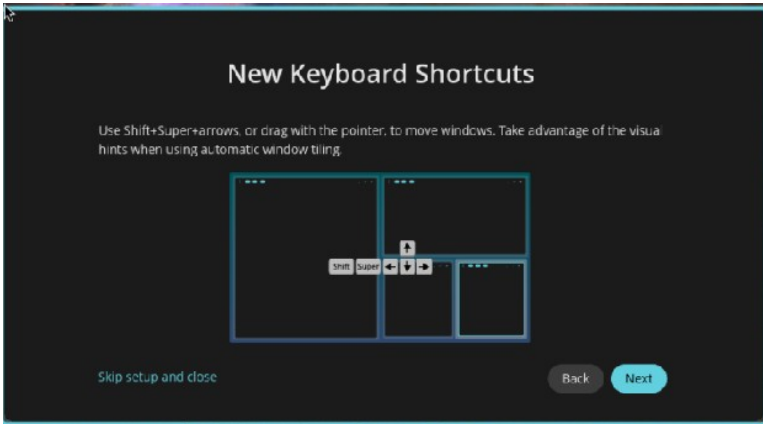
Screenshot by Don Watkins CC-by-SA 4.0

Choose whether your windows will float or tile.



Screenshot by Don Watkins CC-by-SA 4.0

Then, an introduction to keyboard shortcuts to make window management easier.



Screenshot by Don Watkins CC-by-SA 4.0

The COSMIC desktop represents a significant shift in the way we interact with Linux environments. This desktop environment is not only visually appealing but also highly functional, streamlining workflows and enhancing user experience. I find myself increasingly tempted to make it the primary interface for my System76 Meerkat, as its sleek design and innovative features are making a strong case for a permanent switch. The thoughtful design and user-centric approach make COSMIC a compelling option for anyone looking to elevate their Linux experience.

What is Linux?

David Both
July 14, 2025

Linux is the best-known and most-used open-source operating system. As an operating system, Linux is a software that sits underneath all of the other software on a computer, receiving requests from those programs and relaying these requests to the computer's hardware.

How Does Linux Differ From Other Operating Systems?

In many ways, Linux is similar to other operating systems you may have used before, such as Windows, macOS (formerly OS X), or iOS. Like other operating systems, Linux has a graphical interface and the same types of software you are accustomed to, such as word processors, photo editors, video editors, and so on. In many cases, a software's creator may have made a

Linux version of the same program you use on other systems. In short: if you can use a computer or other electronic device, you can use Linux.

But Linux is also different from other operating systems in many important ways. First, and perhaps most importantly, Linux is open-source software. The code used to create Linux is free and available to the public to view, edit, and—for users with the appropriate skills—to contribute to.

Linux is also different in that, although the core pieces of the Linux operating system are generally common, there are many *distributions* of Linux, which include different software options. This means that Linux is incredibly customizable, because not just applications, such as word processors and web browsers, can be swapped out. Linux users also can choose core components, such as which system displays graphics and other user-interface components.

Who Uses Linux?

You probably already use Linux, whether you know it or not. Depending on which user survey you look at, between one- and two-thirds of the webpages on the Internet are generated by servers running Linux.

Companies and individuals choose Linux for their servers because it's secure, flexible, and you can receive excellent sup-

port from a large community of users, in addition to companies like Canonical, SUSE, and Red Hat, each of which offers commercial support.

Many devices you probably own, such as Android phones and tablets and Chromebooks, digital storage devices, personal video recorders, cameras, wearables, and more, also run Linux. Even your car has Linux running under the hood. Microsoft Windows features Linux components as part of the Windows Subsystem for Linux (WSL).

Who “Owns” Linux?

By virtue of its open-source licensing, Linux is freely available to anyone. However, the trademark on the name “Linux” rests with its creator, Linus Torvalds. The source code for Linux is under copyright by its many individual authors and licensed under the GPLv2 license.

The term “Linux” technically refers to just the Linux kernel. Most people refer to the entire operating system as “Linux” because, to most users, an OS includes a bundle of programs, tools, and services (like a desktop, clock, an application menu, and so on). Some people, particularly members of the Free Software Foundation [<https://www.fsf.org/>], refer to this collection as GNU/Linux, because many vital tools included are GNU components. However, not all Linux installations use

GNU components as a part of the operating system: Android, for example, uses a Linux kernel but relies very little on GNU tools.

What is the Difference Between Unix and Linux?

You may have heard of Unix, which is an operating system developed in the 1970s at Bell Labs by Ken Thompson, Dennis Ritchie, and others. Unix and Linux are similar in many ways, and in fact, Linux was originally created to be indistinguishable from Unix. Both have similar tools for interfacing with the system, programming tools, filesystem layouts, and other key components. However, not all Unices are free and open source.

Over the years, a number of different operating systems have been created that attempted to be “unix-like” or “unix-compatible,” but Linux has been the most successful, far surpassing its predecessors in popularity.

How Was Linux Created?

Linux was created in 1991 by Linus Torvalds, a then-student at the University of Helsinki. Torvalds built Linux as a free and

open-source alternative to Minix, another Unix clone that was predominantly used in academic settings. He originally intended to name it “Freax,” but the administrator of the server Torvalds used to distribute the original code named his directory “Linux” after a combination of Torvalds’ first name and the word Unix, and the name stuck.

How Can I Get Started Using Linux?

There’s some chance you’re using Linux already and don’t know it, but if you’d like to install Linux on your home computer to try it out, the easiest way is to pick a popular distribution designed for your platform (for example, laptop or tablet device) and give it a try. Although there are numerous distributions available, most of the older, well-known distributions are good choices for beginners because they have large user communities that can help answer questions if you get stuck or can’t figure things out. Popular Linux distributions include Elementary OS [<https://elementary.io/>], Fedora [<https://fedoraproject.org/>], Mint [<https://linuxmint.com/>], and Ubuntu [<https://ubuntu.com/>] but there are many others. It’s a common saying that the best Linux distro is the one that works best on your computer, so try a few to see which one best suits your hardware and your style of working.

You can install Linux on your current computer (be sure to back up your data first), or you can buy a System76

[<https://system76.com/>] or Purism [<https://puri.sm/>] computer with Linux already installed. If you're not looking for the fastest computing experience possible, you can also install Linux on old computers or buy a Raspberry Pi.

Ultimately, getting started with Linux is a matter of *getting started with Linux*. The sooner you try it, the sooner you'll get comfortable with it, and eventually you'll blissfully forget that non-open operating systems exist!

How Can I Contribute to Linux?

Most of the Linux kernel is written in the C programming language, with a little bit of assembly and other languages sprinkled in. If you're interested in writing code for the Linux kernel itself, a good place to get started is going to this external website called the Kernel Newbies, specifically the FAQ section, which will explain some of the concepts and processes you'll want to be familiar with.

But the Linux community is much more than the kernel and needs contributions from lots of other people besides programmers. Every distribution contains hundreds or thousands of programs that can be distributed along with it, and each of these programs, as well as the distribution itself, needs a variety of people and skill sets to make them successful, including:

- Testers: to make sure everything works on different configurations of hardware and software, and to report the bugs when it does not.
- Designers: to create user interfaces and graphics distributed with various programs.
- Writers: those who can create documentation, how-tos, and other important text distributed with software.
- Translators: to take programs and documentation from their native languages and make them accessible to people around the world.
- Packagers: to take software programs and put all the parts together to make sure they run flawlessly in different distributions.
- Enthusiasts: to spread the word about Linux and open source in general.
- And of course, developers to write the software itself.

Where can I Learn More About Linux?

Both.org has a large archive of Linux-related articles. To view our Both.org archive, browse our Linux tag.

This article was originally published on [Opensource.com](https://opensource.com) under a [CC-by-SA 4.0 license](https://creativecommons.org/licenses/by-sa/4.0/). Modifications by Both.org.

Why I Switched to Linux

Seth Kenlon

August 27, 2025

When I first encountered Linux as an operating system, it wasn't exactly love at first sight. As a kid, I learned computing on an Apple II clone called the LASER 128 and eventually an Apple II GS (which I happily ran until 1999). Then as now, when I commit to a computer, it's for the long term. I enjoy the process of becoming familiar with the software and hardware I run on a computer, so when I discovered Linux, my immediate reaction was frustration. It wasn't enough like what I was used to; it made me feel like I had to start learning yet another computer all over again. The thing I never expected was that this very discomfort turned out to be the best thing for me.

Being an Apple II user in an "IBM clone" world meant I was an expert on a very niche topic. For most of my life, that was a useless qualification, aside from sometimes getting to skip gym class so I could fix computer problems for my overworked school teachers.

All Good Computer Stories Start with Printers

As an adult, knowing all about Apple software and hardware meant I could easily contract myself to publishing houses in need of help setting up printers (the publishing industry used Macintoshes running a proprietary software called Quark Xpress).

It was easy money for a while, until Mac OS 9 was discontinued and Mac OS X became the default. I was working at a computer store at the time, so I got a employee discount on a display Macintosh computer, and already I was hearing a lot about how the new operating system was built on UNIX. My father had been running Mac OS Server for years on his home network, so I had a concept of what UNIX was but no experience with it. I was curious, so I started to read about it.

With almost no printer drivers available for the operating system, people were desperate for a solution, and not by coincidence, I had just recently learned about the Common UNIX Printing System (CUPS). To me, it was a simple trick you did in a web browser. To everyone else, it was magic (including to Apple itself, which purchased CUPS wholesale not long afterwards).

I was providing a legitimate service to people, but I was keenly aware that I was benefiting from something called *open-source* software. It wasn't really me making people's printers work, but because I had a little extra knowledge, people were actually depending on me. Following this realization

through to its logical conclusion, that meant that with even more knowledge about UNIX and the world of open-source software, I would be able to depend solely on myself. I'd be immune to the turmoil I was witnessing in the world of Apple and had witnessed during Microsoft's Windows Me misstep (the short-lived OS before Windows XP).

First Steps with UNIX

My first steps with UNIX were on a Macintosh computer. I'd bought the Visual Quickstart Guide to UNIX by Deborah S. Ray and Eric J. Ray. I was opening the Terminal application as often as I possibly could just for the chance to learn yet another command. I discovered the Fink Project [<https://www.finkproject.org/>] and then later the Mac Ports [<https://www.macports.org/>] repository and started compiling UNIX software on my Macintosh.

I discovered software that freed me from trying to use software I couldn't afford and started using applications like OpenOffice and GIMP. The applications were different, but I could see that learning new software meant I wouldn't ever have to worry about acquiring software I didn't like and couldn't afford to run legitimately.

I still had a problem, though. I was running my applications on an operating system I didn't really trust. Apple had

changed a lot about its OS over a short time, and I didn't always enjoy their choices. I enjoyed even less that I couldn't opt out of them or choose not to use some of the "features." Discovering "secret" UNIX software emboldened me to find more open-source applications, and I eventually cobbled together a collection of workarounds to make up for annoyances in the OS, but deep down I could sense that I'd reached my tolerance threshold for an operating system that was swiftly trying to push me down an upgrade path I neither wanted nor could afford.

My Initial Linux Experience

Having read one book about UNIX, I started reading in tech and film magazines about something called Linux. I didn't understand what it was, because to my mind, a computer and its operating system were inseparable. When I went to a store and bought a computer, I was buying the hardware and that graphical desktop thing I saw when I pressed the power button. I couldn't conceive how you could acquire a disc, put it into a computer, boot off the disc, and then install a new operating system. But I found a book that included a Mandriva Linux install disc in the back, and when a client told me to keep the PC they were upgrading from, I took it home to finally try Linux.

With nothing but skepticism, I inserted the Mandriva disc into my new PC, held down the key that took me to BIOS, and booted off the disc. I fully expected to see Windows because I still couldn't understand how a PC could possibly be anything but a Windows machine, but instead I got the shining star logo of Mandriva Linux (now OpenMandriva [<https://www.openmandriva.org/>] and Mageia Linux [<https://www.mageia.org/en/>]).

I was using Linux!

But Linux wasn't modeled after Mac OS or Mac OS X, the operating systems that had been, for my entire life, synonymous with *computing*. Linux was different. To my advantage, many applications were already familiar to me because I'd been learning them in my Macintosh's UNIX environment. However, the desktop itself was totally foreign. I spent months trying different desktops, with a focus on the ones that people said were "more like Mac OS" than others. I don't think the people who said that had ever used a Mac, because there just wasn't a drop-in replacement. Even on desktops that had been meticulously themed to look like Mac OS's Finder didn't act like Mac OS's Finder. There were too many options here and too few options there. Nothing ever responded the way I expected it to.

The Power of Distraction

Luckily, as frustrated as I was with the differences between Mac OS and Linux desktops, I was equally intrigued by the differences in the operating systems. I loved that I could change the way I interacted with my computer just by adding some lines to a configuration file. I loved how Linux allowed me to build commands that could operate on hundreds of files and then run them in the background while I worked on something entirely different. I'd gained an avatar of myself that could perform routine tasks for me while I worked on new challenges.

I loved that I could configure my computer to use applications of my own choice for major tasks. I got to choose my desktop, my file manager, my network manager, my video player (no more Quicktime!), the way I routed audio, my photo library, and practically anything else I wanted to customize. There were no underhanded hacks required, nothing that would get reset by an update, nothing that I needed to keep from phoning home, and nobody telling me how I had to use my computer.

Curing the Not-Invented-Here Syndrome

It didn't take long for me to feel empowered to just start fixing my "problems" myself. Some things I didn't know how to do.

For example, Mac OS famously had (and maybe still has) a “global” menu bar that changes depending on which application you happen to have in focus. At the time, I thought this was reasonable because I couldn’t imagine wanting to see more than one application’s menu bar. I had to give that up, or at least de-prioritize it.

Other things I could replicate. For example, most keyboard shortcuts for Linux desktops were unfamiliar to me, so I methodically reconfigured the functions I was used to on Mac OS with the keyboard shortcuts I knew.

I didn’t know it at the time, but I was using one of Linux’s most powerful features for the most regressive of tasks. I had fallen prey to the not-invented-here syndrome. I was so inflexible in how I used computers that I was spending all my energy on “fixing” problems that weren’t problems. Luckily, I also fell prey to that other reviled syndrome of complacency. The more I used Linux, the more comfortable I became with the way its desktop worked. The more I used GIMP, the more comfortable I became with what it did differently from other photo editors.

What I didn’t expect was to prefer those differences, but that’s exactly what happened.

I still remember the day, about two years after switching to Linux, when I sat in front of a friend’s Mac. I took it for granted that I knew how Mac OS worked. After all, I’d spent the past two years complaining about the ways Linux wasn’t

like Mac OS. Imagine my surprise when I got frustrated that I couldn't have two applications open on my screen and see both of their menu bars. What I could do in a glance on Linux required a move and a click on Mac.

The keyboard shortcuts didn't work the way I remembered, either. There were almost no global shortcuts. On Linux, I used the Super key for global shortcuts and Control and Alt for application-specific shortcuts. But on Mac OS, I couldn't use a shortcut for Finder while another application was in focus.

Everything else (binary configurations, preferences you didn't actually control or own, and so on) I'd already happily abandoned. As I floundered at my surprising unfamiliarity with the OS I'd grown up with, it occurred to me that it was often easier to learn something new than to force complacency for a familiar but outdated system.

Continuing Education

Linux has taught me a lot, and not just about computing. I learned to code on Linux and how to manage servers; what started out as a hobby has grown into a fulfilling career. Most importantly, I learned the value of evaluating problems and prioritizing solutions. Switching to Linux was a response to the problematic Mac operating system, but it solved more than just proprietary lock-in. Learning Linux gave me the flexibility

to interface with computers on my own terms, and it taught me flexibility in how I approached any given task.

Of course, learning Linux isn't something I've completed. As with most things in life, Linux is developing and changing. There's always something new to learn, and embracing that can lead to new discoveries about computing and, in some cases, about yourself.

