



Home / Innovation / Cloud

Without Dennis Ritchie, there would be no Jobs

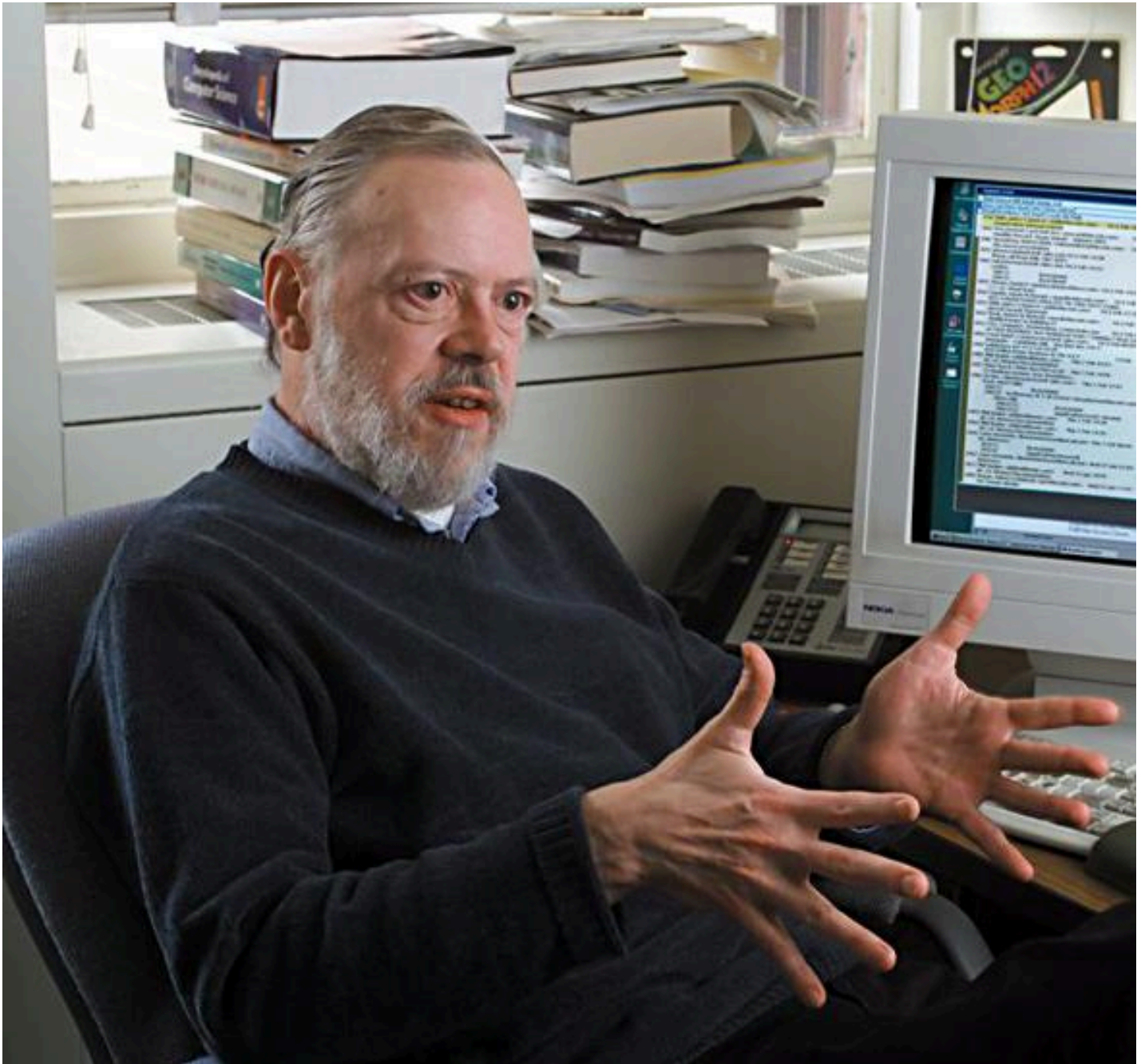
Modern computing as we know it would not have existed without Dennis Ritchie. Remembering the Father of C and UNIX on what would have been his 80th birthday.



Written by **Jason Perlow**, Senior Contributing Writer

Sept. 9, 2022 at 1:29 p.m. PT





Dennis Ritchie

AT&T Bell UNIX Systems Laboratories

Editor's note: This article was originally published in 2011 and updated in 2022.

Eleven years ago next month, we lost two industry giants. One of them would have been 80 years old today.

It is undeniable that Steve Jobs brought us innovation and iconic products like the world had never seen, as well as a cult following of consumers and end users that mythologized him.





Without his contributions, it's likely none of us would be using personal computers today, sophisticated software applications, or even the modern Internet.

The likes of which will probably never be seen again.

At the time of Jobs' passing, I paid my respects and acknowledged his influence, like many in this industry, despite my documented differences with the man and his company,

But the "magical" products that Apple and Steve Jobs -- as well as many other companies created owe just about everything we know and write about in modern computing as it exists today to Dennis Ritchie, who passed away on October 12, 2011, at the age of 70.

Dennis Ritchie?

The younger generation that reads this column is probably scratching their heads. Who was Dennis Ritchie?

Dennis Ritchie wasn't some meticulous billionaire wunderkind from Silicon Valley that mystified audiences with standing-room-only presentations in his minimalist black mock turtleneck with new shiny products and wild rhetoric aimed against his competitors.

No, Dennis Ritchie was a bearded, somewhat disheveled computer scientist who wore cardigan sweaters and had a messy office.

Unlike Jobs, a college dropout, he was a Ph.D., a Harvard University grad with physics and applied mathematics degrees.

And instead of the gleaming Silicon Valley, he worked at AT&T Bell Laboratories in New Jersey.

Yes, Jersey. As in "What exit?"

/ ZDNET recommends

The best programming languages

Here's a list of the most popular programming languages and where to learn them

Read now →

Steve Jobs has frequently been compared to Thomas Edison for the quirkiness of his personality and inventive nature.

I have my issues with that comparison in that we are giving Jobs credit for being an actual technologist and someone who invented something.

It is important to realize that while the man was brilliant in his own way and his contributions were extremely important to the technology and computer industries, Steve Jobs was not a technologist.

Indeed, he had a very strong sense of style and industrial design, understood customers' wants, and was a master marketer and salesperson. All of these make him a giant in our industry.

But inventor? No.

Dennis M. Ritchie, on the other hand, invented and co-invented two key software technologies that effectively make up the DNA of every single computer software product we use directly or indirectly in the modern age.

It sounds like a wild claim, but it really is true.

First, let's start with the C programming language.

C, the basis of all modern programming languages

Developed by Ritchie between 1969 and 1973, C is considered to be the first truly modern and portable programming language. In the 53 years since its introduction, it has been ported to practically every systems architecture and operating system in existence.

Because it is an imperative, compiled, procedural programming language, allowing for lexical variable scope and recursion, and allowing low-level access to memory, and complex functionality for I/O and string manipulation, the language became quite versatile.

This allowed Ritchie and Brian Kernighan to refine it to a degree that eventually was further refined by the X3J11 committee of the American National Standards Institute as the ANSI C programming language in 1989.

In 1978, Kernighan and Ritchie published the book *"The C Programming Language"*. Referred to by many simply as "K&R," it is considered a computer science masterpiece and a critical reference for explaining modern programming concepts. It is still used as a text when teaching programming to students in computer science curricula even today.

ANSI C as a programming language is still used heavily today, and it has since mutated into several sister languages, all of which have strong followings.

/ see also

How to replace Windows with Linux Mint on your PC

5 best Linux commands for troubleshooting problems (and how I use them)

5 reasons why Pop!_OS is this Linux pro's favorite distro

5 best open-source email clients for Linux (and why Geary is my go-to)

The most popular, C++ (pronounced "C plus plus"), which was introduced by [Bjarne Stroustrup](#) in 1985 and added support for object-oriented programming and classes, is used on a variety of operating systems, including every major UNIX derivative, including Linux and the Mac, and is the primary programming language that has been used for Microsoft Windows software development for at least 20 years.

[Objective-C](#), created by Brad Cox and Todd Love in the 1980s at a company called [Stepstone](#), added [Smalltalk](#) messaging capabilities to the language, further extending the language's object-oriented and code re-usability features.

It was primarily considered an obscure derivative of C until it was popularized in the [NeXTStep](#) and [OpenStep](#) operating systems in the late 1980s and early 1990s on Steve Jobs's [NeXT](#) computer systems, the company he formed after Apple's board ousted him in 1985.

What happened "next," of course, is computing history. Apple purchased NeXT in 1996, and Jobs returned to become CEO of the company in 1997.

In 2001, Apple launched Mac OS X, which heavily uses Objective-C and object-oriented technologies introduced in NeXTStep/OpenStep.

While C++ is also used heavily on the Mac, Objective-C is used to program to the native object-oriented "[Cocoa](#)" API in the [XCode IDE](#), which is central to the gesture recognition and animation features on iOS that powers the iPhone and the iPad.

Objective-C also provides frameworks for the [Foundation Kit](#) and [Application Kit](#) that are essential to building native OS X and iOS applications.

Microsoft has its derivative of C in [C#](#) (pronounced "C Sharp"), which was introduced in 2001 and served as the foundation for programming within the .NET framework.

C# is also the basis for programming the new Modern applications in the Windows Runtime (WinRT) that evolved into the Universal Windows Platform (UWP) in Windows 10. It is also used within Linux and other Unix derivatives as Mono's programmatic environment, a portable version of the .NET framework.

But C's influence doesn't end at C language derivatives. Java, an important enterprise programming language (and has morphed into Dalvik and the Android Runtime, which is used as the primary programming environment for Android) is heavily based on C syntax.

Other languages such as Ruby, Perl, and PHP which form the basis for the modern dynamic Web, all use syntax introduced in C, created by Dennis Ritchie.

/ featured

The Microsoft 365 Copilot launch was a total disaster

Your TV's USB port has superpowers: 4 useful benefits you're not taking advantage of

Galaxy S25 Ultra vs. iPhone 16 Pro Max: We tested both, and the winner is not so obvious

If you're ready to break up with Windows, this is the Linux distro I suggest for new users

So it could be said that without the work of Dennis Ritchie, we would have no modern software... at all.

I could end this article simply with what Ritchie's development of C means to modern computing and how it impacts everyone. But I would only really be describing half of a life's work of this man.

UNIX: The operating system that began it all

Ritchie is also the co-creator of the UNIX operating system. Of course, after being prototyped in assembly language, it was completely rewritten in the early 1970s in C.

Since the first implementation of "Unics" booted on a DEC PDP-7 back in 1969, it has mutated into many other similar operating systems running on various systems architectures.

Name a major computer vendor, and every single one has had an implementation of UNIX at some time. Even Microsoft, which once owned a product called XENIX and has since sold it to SCO (now defunct).

You'll want to click and zoom into this picture so you can get a better understanding of this "family."

Essentially, there are three main branches.

One branch is the "System V" UNIXes that we know today as IBM AIX, Oracle Solaris, and Hewlett Packard's HP-UX. All of these are considered to be "Big Iron" OSes that drive critical transaction-oriented business applications and databases in the largest enterprises in the world, the Fortune 1000.

Without the System V UNIXes, the Fortune 1000 probably wouldn't get much of anything done. Business would essentially grind to a halt.

They may only represent about 10 to 20 percent of any particular enterprise's computing population, but it's a very important 20 percent.

The second branch, the BSDs (Berkeley Systems Distribution) includes FreeBSD/NetBSD/OpenBSD, which form the basis for both Mac OS X and the iOS that powers the iPhone and iPad. They are also the backbone supporting much of the critical infrastructure that runs the Internet.

The third branch of UNIX is not even a branch at all -- GNU/Linux. The Linux kernel (developed by Linus Torvalds) combined with the GNU user-space programs, tools, and utilities provides for a complete re-implementation of a "UNIX-like" or "UNIX-compatible" operating system from the ground up.

Linux, of course, has become the most disruptive of all the UNIX operating systems. It scales from the tiny, from embedded microcontrollers to smartphones to tablets, desktops, and even the most powerful supercomputers.

One such Linux supercomputer, IBM's Watson, beat Ken Jennings on *Jeopardy!* While the world watched in awe.

Still, it is essential to recognize that Linux and GNU contain no UNIX code, hence the Free Software recursive phrase "GNU's Not UNIX."

But by design, GNU/Linux behaves much like UNIX, and it could be said that without UNIX being developed by Ritchie and his colleagues Brian Kernighan, Ken Thompson, Douglas McIlroy, and Joe Ossanna at Bell Labs in the first place, there never would have been any Linux or an Open Source Software movement.

Or a Free Software Foundation or a Richard Stallman to be glad Steve Jobs is gone, for that matter.

But enough of religion and ideology. We owe much to Dennis Ritchie, more than we can ever imagine. Without his contributions, it's likely none of us would be using personal computers today, sophisticated software applications, or even the modern Internet.

No streaming devices and no Macs, iPhones, iPads, and Watches for Apple to make Amazingly Great. No Microsoft Windows 11 or Surface Books. No Androids, or Chromebooks. No Alexa, no Netflix.

No Cloud, no AWS, no Azure.

No "Apps for That." *No Internet of Anything.*

To Dennis Ritchie, I thank you -- for giving all of us the technology to be the technologists we are today.

Dennis Ritchie (standing) and Ken Thompson with a PDP-11, circa 1972 (Source: [Dennis Ritchie homepage](#))

 [Editorial standards](#)

[show comments](#) ↓

**we equip you to harness the power of
disruptive innovation, at work and at home.**

[topics](#)

[galleries](#)

[videos](#)

[do not sell or share my personal information](#)

[about ZDNET](#)

[meet the team](#)

[sitemap](#)

[reprint policy](#)

[join | log in](#)

newsletters

site assistance

licensing

accessibility

© 2025 ZDNET, A Ziff Davis company. All rights reserved. **Privacy Policy** ||| **Advertise** | **Terms of Use**