Emacs

Emacs /ˈiːmæks/ or **EMACS** (**Editor MACroS**)[3][4][5] is a family of text editors that are characterized by their extensibility.[6] The manual for the most widely used variant,[7] GNU Emacs, describes it as "the extensible, customizable, self-documenting, real-time display editor".[8] Development of the first Emacs began in the mid-1970s, and work on its direct descendant, GNU Emacs, continues actively as of 2020.

Emacs has over 10,000 built-in commands and its user interface allows the user to combine these commands into macros to automate work. Implementations of Emacs typically feature a dialect of the Lisp programming language that provides a deep extension capability, allowing users and developers to write new commands and applications for the editor. Extensions have been written to manage email, files, outlines, and RSS feeds,[9] as well as clones of *ELIZA*, *Pong*, *Conway's Life*, *Snake* and *Tetris*.[10]

The original EMACS was written in 1976 by David A. Moon and Guy L. Steele Jr. as a set of Editor MACroS for the TECO editor.[2][3][4][5][11] It was inspired by the ideas of the TECO-macro editors TECMAC and TMACS.[12]

The most popular, and most ported, version of Emacs is GNU Emacs, which was created by Richard Stallman for the GNU Project.[13] XEmacs is a variant that branched from GNU Emacs in 1991. GNU Emacs and XEmacs use similar Lisp dialects and are, for the most part, compatible with each other. XEmacs development is inactive.

Emacs is, along with vi, one of the two main contenders in the traditional editor wars of Unix culture. Emacs is among the oldest free and open source projects still under development.[14]

Contents

Emacs

Editing multiple Dired buffers in GNU Emacs

Original author(s)	David A. Moon, Guy L. Steele Jr.
Developer(s)	Various free/libre software developers, including volunteers and commercial developers
Initial release	1976[1][2]
Stable release	27.1 (August 10, 2020) [±] (https://en.wikipedia.org/w/index.php?title=Template:Latest_stable_software_release/Emacs&action=edit)
Written in	Lisp, C
Operating system	Cross-platform
Type	Text editor

History

Emacs was started by Guy L. Steele Jr. as a project to unify the many divergent TECO command sets and key bindings at MIT[4]

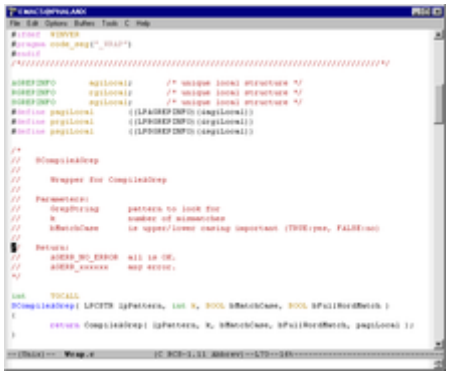
The interface of Emacs was influenced by the design of the Symbolics space-cadet keyboard[15]

Emacs development began during the 1970s at the MIT AI Lab, whose PDP-6 and PDP-10 computers used the Incompatible Timesharing System (ITS) operating system that featured a default line editor known as Tape Editor and Corrector (TECO). Unlike most modern text editors, TECO used separate modes in which the user would either add text, edit existing text, or display the document. One could not place characters directly into a document by typing them into TECO, but would instead enter a character ('i') in the TECO command language telling it to switch to input mode, enter the required characters, during which time the edited text was not displayed on the screen, and finally enter a character (<esc>) to switch the editor back to command mode. (A similar technique was used to allow overtyping.) This behavior is similar to that of the program ed.

Richard Stallman visited the Stanford AI Lab in 1972 or 1974 and saw the lab's *E* editor, written by Fred Wright.[16] He was impressed by the editor's intuitive WYSIWYG (What You See Is What You Get) behavior, which has since become the default behavior of most modern text editors. He returned to MIT where Carl Mikkelsen, a hacker at the AI Lab, had added to TECO a combined display/editing mode called *Control-R* that allowed the screen display to be updated each time the user entered a keystroke. Stallman reimplemented this mode to run efficiently and then added a macro feature to the TECO display-editing mode that allowed the user to redefine any keystroke to run a TECO program.[5]

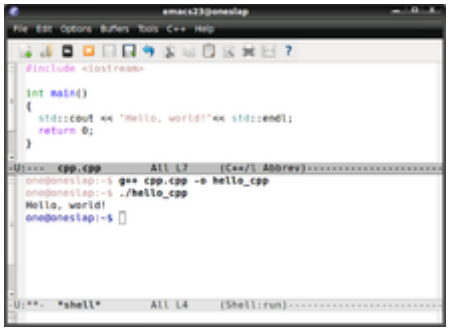
E had another feature that TECO lacked: random-access editing. TECO was a page-sequential editor that was designed for editing paper tape on the PDP-1 and typically allowed editing on only one page at a time, in the order of the pages in the file. Instead of adopting E's approach of structuring the file for page-random access on disk, Stallman modified TECO to handle large

buffers more efficiently and changed its file-management method to read, edit, and write the entire file as a single buffer. Almost all modern editors use this approach.



Editing C source code in GNU Emacs

The new version of TECO quickly became popular at the AI Lab and soon accumulated a large collection of custom macros whose names often ended in *MAC* or *MACS*, which stood for *macro*. Two years later, Guy Steele took on the project of unifying the diverse macros into a single set.[17] Steele and Stallman's finished implementation included facilities for extending and documenting the new macro set.[5] The resulting system was called EMACS, which stood for *Editing MACroS* or, alternatively, *E with MACroS*. Stallman picked the name Emacs "because <E> was not in use as an abbreviation on ITS at the time."[18] An apocryphal hacker koan alleges that the program was named after *Emack & Bolio's*, a popular Cambridge ice cream store.[19] The first operational EMACS system existed in late 1976.[20]



Editing, compiling and executing C++ code from GNU Emacs

Stallman saw a problem in too much customization and *de facto* forking and set certain conditions for usage. He later wrote:[20]

> EMACS was distributed on a basis of communal sharing, which means all improvements must be given back to me to be incorporated and distributed.

The original Emacs, like TECO, ran only on the PDP-10 running ITS. Its behavior was sufficiently different from that of TECO that it could be considered a text editor in its own right, and it quickly became the standard editing program on ITS. Mike McMahon ported Emacs from ITS to the TENEX and TOPS-20 operating systems. Other contributors to early versions of Emacs include Kent Pitman, Earl Killian, and Eugene Ciccarelli. By 1979, Emacs was the main editor used in MIT's AI lab and its Laboratory for Computer Science.[21]

Implementations

Early implementations

In the following years, programmers wrote a variety of Emacs-like editors for other computer systems. These included EINE (*EINE Is Not EMACS*) and ZWEI[22] (*ZWEI Was EINE Initially*), which were written for the Lisp machine by Mike McMahon and Daniel Weinreb, and Sine (*Sine Is Not Eine*),[23] which was written by Owen Theodore Anderson. Weinreb's EINE was the first Emacs written in Lisp. In 1978, Bernard Greenberg wrote Multics Emacs almost entirely in Multics Lisp at Honeywell's Cambridge Information Systems Lab. Multics Emacs was later maintained by Richard Soley, who went on to develop the NILE Emacs-like editor for the NIL Project, and by Barry Margolin. Many versions of Emacs, including GNU Emacs, would later adopt Lisp as an extension language.

James Gosling, who would later invent NeWS and the Java programming language, wrote Gosling Emacs in 1981. The first Emacs-like editor to run on Unix, Gosling Emacs was written in C and used Mocklisp, a language with Lisp-like syntax, as an extension language.

Early Ads for Computer Corporation of America's *CCA EMACS* (Steve Zimmerman).[24] appeared in 1984.[25] 1985 comparisons to GNU Emacs, when it came out, mentioned free vs. $2,400.[26]

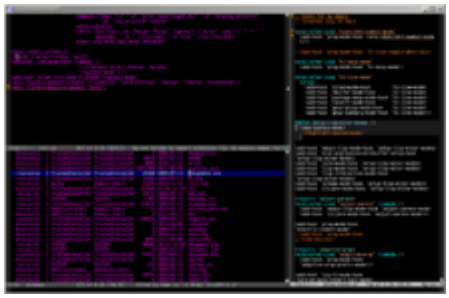
James Gosling wrote the first Emacs-like editor to run on Unix (Gosling Emacs) in 1981

GNU Emacs

Richard Stallman began work on GNU Emacs in 1984 to produce a free software alternative to the proprietary Gosling Emacs. GNU Emacs was initially based on Gosling Emacs, but Stallman's replacement of its Mocklisp interpreter with a true Lisp interpreter required that nearly all of its code be rewritten. This became the first program released by the nascent GNU Project. GNU Emacs is written in C and provides Emacs Lisp, also implemented in C, as an extension language. Version 13, the first public release, was made on March 20, 1985. The first widely distributed version of GNU Emacs was version 15.34, released later in 1985. Early versions of GNU Emacs were numbered as *1.x.x*, with the initial digit denoting the version of the C core. The *1* was dropped after version 1.12, as it was thought that the major number would never change, and thus the numbering skipped from *1* to *13*.[27] In September 2014, it was announced on the GNU emacs-devel mailing list that GNU Emacs would adopt a rapid release strategy and version numbers would increment more quickly in the future.[28]

GNU Emacs running in a text console

GNU Emacs was later ported to Unix. It offered more features than Gosling Emacs, in particular a full-featured Lisp as its extension language, and soon replaced Gosling Emacs as the *de facto* Unix Emacs editor. Markus Hess exploited a security flaw in GNU Emacs' email subsystem in his 1986 cracking spree in which he gained superuser access to Unix computers.[29]



GNU Emacs running on Microsoft Windows

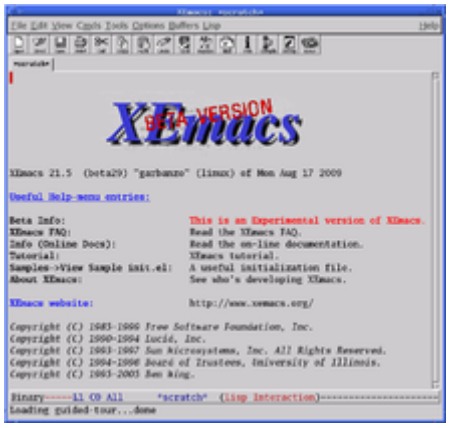
Most of GNU Emacs functionality is implemented through a scripting language called Emacs Lisp. Because about 70% of GNU Emacs is written in the Elisp extension language,[30] one only needs to port the C core which implements the Elisp interpreter. This makes porting Emacs to a new platform considerably less difficult than porting an equivalent project consisting of native code only.

GNU Emacs development was relatively closed until 1999 and was used as an example of the *Cathedral* development style in *The Cathedral and the Bazaar*. The project has since adopted a public development mailing list and anonymous CVS access. Development took place in a single CVS trunk until 2008 and was then switched to the Bazaar DVCS. On November 11, 2014, development was moved to Git.[31]

Richard Stallman has remained the principal maintainer of GNU Emacs, but he has stepped back from the role at times. Stefan Monnier and Chong Yidong were maintainers from 2008 to 2015.[32][33] John Wiegley was named maintainer in 2015 after a meeting with Stallman at MIT.[34] As of early 2014, GNU Emacs has had 579 individual committers throughout its history.[35]

XEmacs

Lucid Emacs, based on an early alpha version of GNU Emacs 19, was developed beginning in 1991 by Jamie Zawinski and others at Lucid Inc. One of the best-known early forks in free software development occurred when the codebases of the two Emacs versions diverged and the separate development teams ceased efforts to merge them back into a single program.[36] Lucid Emacs has since been renamed XEmacs. Its development is currently inactive, with the most recent stable version 21.4.22 released in January 2009 (while a beta was released in 2013), while GNU Emacs has implemented many formerly XEmacs-only features.[37]



XEmacs 21.5 on GNU/Linux

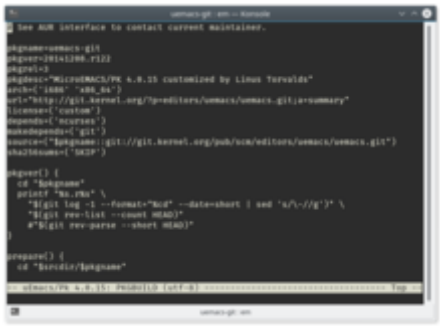
Other forks of GNU Emacs

Other notable forks include:

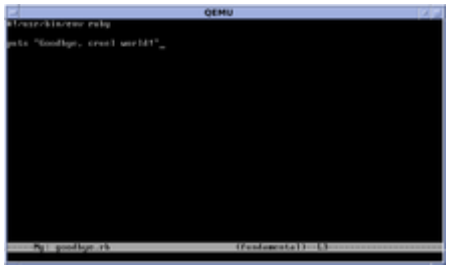
- Aquamacs – based on GNU Emacs (Aquamacs 3.2 is based on GNU Emacs version 24 and Aquamacs 3.3 is based on GNU Emacs version 25) which focuses on integrating with the Apple Macintosh user interface
- Meadow – a Japanese version for Microsoft Windows[38]
- SXEmacs – Steve Youngs' fork of XEmacs[39]

Various Emacs editors

In the past, projects aimed at producing small versions of Emacs proliferated. GNU Emacs was initially targeted at computers with a 32-bit flat address space and at least 1 MiB of RAM.[40] Such computers were high end workstations and minicomputers in the 1980s, and this left a need for smaller reimplementations that would run on common personal computer hardware. Today's computers have more than enough power and capacity to eliminate these restrictions, but small clones have more recently been designed to fit on software installation disks or for use on less capable hardware.[41]


uEmacs/Pk 4.0.15 on Linux

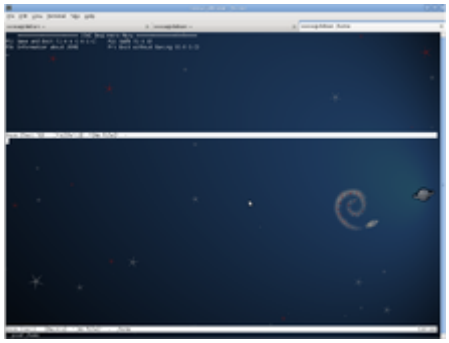
Other projects aim to implement Emacs in a different dialect of Lisp or a different programming language altogether. Although not all are still actively maintained, these clones include:



The *mg* tiny Emacs-like editor in OpenBSD 5.3. Editing Ruby source code

- MicroEMACS, which was originally written by Dave Conroy and further developed by Daniel Lawrence and which exists in many variations.
- mg, originally called MicroGNUEmacs and, later, mg2a, a public-domain offshoot of MicroEMACS intended to more closely resemble GNU Emacs. Now installed by default on OpenBSD.
- JOVE (Jonathan's Own Version of Emacs), Jonathan Payne's non-programmable Emacs implementation for UNIX-like systems.
- MINCE (MINCE Is Not Complete Emacs), a version for CP/M and later DOS, from Mark of the Unicorn. MINCE evolved into Final Word, which eventually became the Borland Sprint word processor.

- Perfect Writer, a CP/M implementation derived from MINCE that was included circa 1982 as the default word processor with the very earliest releases of the Kaypro II and Kaypro IV. It was later provided with the Kaypro 10 as an alternative to WordStar.
- Freemacs, a DOS version that uses an extension language based on text macro expansion and fits within the original 64 KiB flat memory limit.
- Zile. Zile was a recursive acronym for *Zile Is Lossy Emacs*,[42] but the project was rewritten in Lua and now gives the expansion as Zile Implements Lua Editors. The new Zile still includes an implementation of Emacs in Lua called Zemacs. There is also an implementation of vi called Zi.
- Zmacs, for the MIT Lisp Machine and its descendants, implemented in ZetaLisp.
- Climacs, a Zmacs-influenced variant implemented in Common Lisp.
- Epsilon,[43] an Emacs clone by Lugaru Software. Versions for DOS, Windows, Linux, FreeBSD, Mac OS X and O/S 2 are bundled in the release. It uses a non-Lisp extension language with C syntax and used a very early concurrent command shell buffer implementation under the single-tasking MS-DOS.



JOVE running in a Debian box

Zmacs, an Emacs for Lisp machines

- PceEmacs is the Emacs-based editor for SWI-Prolog.
- Amacs, an Apple II ProDOS version of Emacs implemented in 6502 assembly by Brian Fox.[44][45]
- Hemlock, originally written in Spice Lisp, then Common Lisp. A part of CMU Common Lisp. Influenced by Zmacs. Later forked by Lucid Common Lisp (as Helix), LispWorks and Clozure CL projects. There is also a Portable Hemlock project, which aims to provide a Hemlock, which runs on several Common Lisp implementations.
- umacs,[46] an implementation under OS-9
- edwin, an Emacs-like text editor included with MIT/GNU Scheme.

Editors with Emacs emulation

- Joe's Own Editor emulates Emacs keybindings when invoked as `jmacs`.
- JED has an emulation mode for Emacs.
- Eclipse (IDE) provides a set of Emacs keybindings.
- IntelliJ IDEA provides a set of Emacs keybindings.
- Epsilon (text editor) Defaults to Emacs emulation and supports a vi mode.
- The Cocoa text system uses some of the same terminology and understands many Emacs navigation bindings. This is possible because the native UI uses the Command key (equivalent to Super) instead of the Control key.[47]
- GNU Readline is a line editor that understands the standard Emacs navigation keybindings. It also has a vi emulation mode.
- GNOME Builder has an emulation mode for Emacs.
- MATLAB provides Emacs keybindings for its editor.[48]

- Visual Studio Code provides an extension (https://marketplace.visualstudio.com/items?itemName=vscodeemacs.emacs) to emulate Emacs keybindings.

Features

Emacs is primarily a text editor and is designed for manipulating pieces of text, although it is capable of formatting and printing documents like a word processor by interfacing with external programs such as LaTeX, Ghostscript or a web browser. Emacs provides commands to manipulate and differentially display semantic units of text such as words, sentences, paragraphs and source code constructs such as functions. It also features *keyboard macros* for performing user-defined batches of editing commands.

GNU Emacs is a *real-time display* editor, as its edits are displayed onscreen as they occur. This is standard behavior for modern text editors but EMACS was among the earliest to implement this. The alternative is having to issue a distinct command to display text, eg after modifying it. This is done in line editors, such as ed (unix), ED (CP/M), and Edlin (MS-DOS).

General architecture

Almost all of the functionality in Emacs, including basic editing operations such as the insertion of characters into a document, is achieved through functions written in a dialect of the Lisp programming language. The dialect used in GNU Emacs is known as Emacs Lisp (ELisp). The ELisp layer sits atop a stable core of basic services and platform abstraction written in the C programming language. In this Lisp environment, variables and functions can be modified with no need to recompile or restart Emacs.

Emacs operates on data structures called *buffers* containing text with additional attributes; every buffer maintains its own *point* (cursor location) and *mark* (another location, delimiting the selected *region* together with the *point*), the name of the file it is *visiting* (if applicable) and the set of active *modes* (exactly one *major mode* and any number of *minor modes*), which control editor behaviour through variables. Elisp code can be executed interactively through named *commands*, which can be bound to key presses or accessed by name; some commands evaluate arbitrary Elisp code from buffers (e.g. `eval-region` or `eval-buffer`).

Buffers are displayed in *windows*, which are tiled portions of the terminal screen or the GUI window (called a *frame* in Emacs terms; multiple frames are possible). Unless configured otherwise, windows include scroll bars, line numbers, a *header line* at the top (usually displaying the buffer title or filename) and a *mode line* at the bottom (usually listing the active modes and point position of the buffer).

Multiple windows can be opened onto the same buffer, for example to see different parts of a long text, and multiple buffers can share the same text, for example to take advantage of different major modes in a mixed-language file. The mode can also be changed manually as needed with `M-x <mode name>`.

Customizability

- Keystrokes can be recorded into macros and replayed to automate complex, repetitive tasks. This is often done on an ad-hoc basis, with each macro discarded after use, although macros can be saved and invoked later.
- At startup, Emacs executes an Emacs Lisp script named `~/.emacs` (recent versions also look for `~/emacs.el`, `~/.emacs.d/init.el`, and `~/.config/emacs/init.el`;[49] Emacs will execute the first one it finds, ignoring the rest). This personal customization file can be arbitrarily long and complex, but typical content includes:

    - Setting global variables or invoking functions to customize Emacs behaviour, for example `(set-default-coding-systems 'utf-8)`
    - Key bindings to override standard ones and to add shortcuts for commands that the user finds convenient but don't have a key binding by default. Example: `(global-set-key (kbd "C-x C-b") 'ibuffer)`
    - Loading, enabling and initializing extensions (Emacs comes with many extensions, but only a few are loaded by default.)
    - Configuring *event hooks* to run arbitrary code at specific times, for example to automatically recompile source code after saving a buffer (`after-save-hook`)
    - Executing arbitrary files, usually to split an overly long configuration file into manageable and homogeneous parts (`~/.emacs.d/` and `~/elisp/` are traditional locations for these personal scripts)
- The *customize* extension allows the user to set configuration properties such as the color scheme interactively, from within Emacs, in a more user-friendly way than by setting variables in `.emacs`: it offers search, descriptions and help text, multiple choice inputs, reverting to defaults, modification of the running Emacs instance without reloading, and other conveniences similar to the preferences functionality of other programs. The customized values are saved in `.emacs` (or another designated file) automatically.
- *Themes*, affecting the choice of fonts and colours, are defined as elisp files and chosen through the customize extension.

Self-documenting

The first Emacs contained a *help* library that included documentation for every command, variable and internal function. Because of this, Emacs proponents described the software as *self-documenting* in that it presents the user with information on its normal features and its current state. Each function includes a documentation string that is displayed to the user on request, a practice that subsequently spread to programming languages including Lisp, Java, Perl, and Python. This help system can take users to the actual code for each function, whether from a built-in library or an added third-party library.

Emacs also has a built-in tutorial. Emacs displays instructions for performing simple editing commands and invoking the tutorial when it is launched with no file to edit. The tutorial is by Stuart Cracraft and Richard Stallman.

Culture

Church of Emacs

The *Church of Emacs*, formed by Richard Stallman, is a parody religion created for Emacs users.[50] While it refers to vi as the *editor of the beast* (vi-vi-vi being 6-6-6 in Roman numerals), it does not oppose the use of vi; rather, it calls it proprietary software anathema. ("Using a free version of vi is not a sin but a penance."[51]) The Church of Emacs has its own newsgroup, `alt.religion.emacs`,[52] that has posts purporting to support this parody religion. Supporters of vi have created an opposing *Cult of vi*.

Stallman has jokingly referred to himself as *St IGNUcius*, a saint in the Church of Emacs.[53]

Emacs pinky

There is folklore attributing a repetitive strain injury colloquially called *Emacs pinky* to Emacs' strong dependence on modifier keys,[54] although there have not been any studies done to show Emacs causes more such problems than other keyboard-heavy computer programs.

Richard Stallman as St IGNUcius, a saint in the *Church of Emacs*

Users have addressed this through various approaches. Some users recommend simply using the two Control keys on typical PC keyboards like Shift keys while touch typing to avoid overly straining the left pinky, a proper use of the keyboard will reduce the RSI. [55] Software-side methods include:[56]

- Customizing the key layout so that the Control key is transposed with the caps lock key.[57] Similar techniques include defining the caps lock key as an additional Control key or transposing the Control and Meta keys. However, as these keys are still pressed with the same finger, they may still contribute to Emacs pinky.
- Software, such as xwrits or the built-in `type-break-mode` in Emacs, that reminds the user to take regularly scheduled breaks.
- Using the ErgoEmacs keybindings (with minor mode `ergoemacs-mode`).[58]
- Customizing the whole keyboard layout to move statistically frequent Emacs keys to more appropriate places.[59]
- Packages such as `ace-jump-mode`[60] or elisp extensions that provide similar functionality of tiered navigation, first asking for a character then replacing occurrences of the character with access keys for cursor movement.
- `evil-mode`, an advanced Vim emulation layer.
- `god-mode`, which provides an approach similar to vim's with a mode for entering Emacs commands without modifier keys.
- Using customized key layout offered by Spacemacs, a project where `Space` key is used as the main key for initiating control sequences. The project also heavily incorporates both `evil-mode` and `god-mode`.[61]
- StickyKeys, which turns key sequences into key combinations.[62]
- Emacs' built-in *`viper-mode`* that allows use of the vi key layout for basic text editing and the Emacs scheme for more advanced features.[63]
- Giving a dual role to a more-comfortably accessed key such as the Space bar so that it functions as a Control key when pressed in combination with other keys. Ergonomic keyboards or keyboards with a greater number of keys adjacent to the space bar, such as Japanese keyboards, allow thumb control of other modifier keys too like Meta or Shift.[64]
- Using a limited ergonomic subset of keybindings, and accessing other functionality by typing `M-x <command-name>`. M-x itself can also be rebound.
- Driving Emacs through voice input.

Hardware solutions include special keyboards such as Kinesis's Contoured Keyboard, which places the modifier keys where they can easily be operated by the thumb, or the Microsoft Natural keyboard, whose large modifier keys are placed symmetrically on both sides of the keyboard and can be pressed with the palm of the hand.[54] Foot pedals can also be used.

The *Emacs pinky* is a relatively recent development. The Space-cadet keyboard on which Emacs was developed had oversized Control keys that were adjacent to the space bar and were easy to reach with the thumb.[65]

Terminology

The word *emacs* is sometimes pluralized as *emacsen*, by phonetic analogy with boxen and VAXen, referring to different varieties of Emacs.[66]

See also

- Comparison of text editors
- Conkeror
- GNU TeXmacs
- List of text editors
- List of Unix commands
- Integrated development environment

References

1. "Emacs Release Dates" (https://www.emacswiki.org/emacs/EmacsReleaseDates).
2. Zawinski, Jamie (2005-06-21) [1999]. "Emacs Timeline" (https://www.jwz.org/doc/emacs-timeline.html). Retrieved 11 August 2015.
3. Greenberg, Bernard S. (1979). *Multics Emacs: The History, Design and Implementation* (http://www.multicians.org/mepap.html).
4. "GNU Emacs FAQ" (https://www.gnu.org/software/emacs/emacs-faq.html#Origin-of-the-term-Emacs).
5. Adrienne G. Thompson. "MACSimizing TECO" (http://www.codeartnow.com/hacker-art-1/macsimizing-teco).
6. "A Tutorial Introduction to GNU Emacs" (http://www2.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html). "For an editor to be called "emacs" the main requirement is that it be fully extensible with a real programming language, not just a macro language."
7. "GNU Emacs Pocket Reference" (https://ebooks-it.org/1565924967-ebook.htm). "GNU Emacs is the most popular and widespread of the Emacs family of editors."
8. "GNU Emacs Manual" (https://www.gnu.org/software/emacs/manual/html_node/emacs/index.html). FSF. Retrieved 24 November 2012.
9. "Introducing Elfeed, an Emacs Web Feed Reader" (http://nullprogram.com/blog/2013/09/04/).
10. "Amusements" (https://www.gnu.org/software/emacs/manual/html_node/emacs/Amusements.html). "Finally, if you find yourself frustrated, try describing your problems to the famous psychotherapist Eliza. Just do M-x doctor."
11. Richard Stallman is often credited as another co-creator, but as Daniel Weinreb wrote, "The original (TECO-based) Emacs was created and designed by Guy L. Steele Jr. and David Moon. After they had it working, and it had become established as the standard text editor at the AI lab, Stallman took over its maintenance." Moon himself responded "All true, so far as I can remember. But in all fairness I have to say that Stallman greatly improved Emacs after he "liberated" it from Guy and me." See Weinreb, Dan (November 11, 2007), "Rebuttal to Stallman's Story About The Formation of Symbolics and LMI" (https://web.archive.org/web/20090101103828/http://danweinreb.org/blog/rebuttal-to-stallmans-story-about-the-formation-of-symbolics-and-lmi), *Dan Weinreb's blog: software and innovation*, archived from the original (http://danweinreb.org/blog/rebuttal-to-stallmans-story-about-the-formation-of-symbolics-and-lmi) on January 1, 2009.
12. "A history of Emacs" (http://www.xemacs.org/Documentation/21.5/html/internals_3.html). *XEmacs Internals Manual*. 2006-12-11. Retrieved 2007-08-22.

13. Allombert, Bill. "Debian Popularity Contest" (http://popcon.debian.org/main/editors/by_vote). *Editors report*. Debian. Retrieved 22 November 2011.
14. "The 10 oldest, significant open-source programs" (https://www.zdnet.com/article/the-10-oldest-significant-open-source-programs/).
15. Raymond, Eric S.; Cameron, Debra; Rosenblatt, Bill (1996). *Learning GNU Emacs, 2nd Edition* (https://books.google.com/books?id=a_Iea3-w-1kC&pg=PA408&dq=bucky+keyboard#PPA408,M1). Sebastopol, CA: O'Reilly. pp. 408–409. ISBN 1-56592-152-6.
16. Arthur Samuel (March 1980). "Essential E" (http://i.stanford.edu/pub/cstr/reports/cs/tr/80/796/CS-TR-80-796.pdf) (PDF). Retrieved 2020-04-19.
17. "EMACS vs. vi: The endless geek 'holy war' " (https://web.archive.org/web/20141126181620/http://archive09.linux.com/feature/19661). Archived from the original (http://archive09.linux.com/feature/19661) on 2014-11-26. Retrieved 2014-05-30. ""EMACS as such actually started out as a standards project," emails Guy Steele"
18. Stallman, Richard M. (1987). "The EMACS Full-Screen Editor" (http://www.lysator.liu.se/history/garb/txt/87-1-emacs.txt). *GARB* (Maj 1987). Lysator, Linköping University. pp. 8–11. Retrieved 2007-09-14.
19. Reynolds, Craig (1992-02-10). Wiseman, David G. (ed.). "The Emac Bolio Name Koan" (http://www.csd.uwo.ca/staff/magi/personal/humour/Computer_Folklore/The%20Emac%20Bolio%20Name%20Koan.html). *David G. Wiseman: Stories of Computer Folklore*. "A cocky novice once said to Stallman: 'I can guess why the editor is called Emacs, but why is the justifier called Bolio?'. Stallman replied forcefully, Names are but names, Emack & Bolio's is the name of a popular ice cream shop in Boston town. Neither of these men had anything to do with the software.' His question answered, yet unanswered, the novice turned to go, but Stallman called to him, 'Neither Emacs nor Bolio had anything to do with the ice cream shop, either.'"
20. Stallman, Richard (March 26, 1981). *EMACS: The Extensible, Customizable, Self-Documenting, Display Editor* (http://dspace.mit.edu/bitstream/handle/1721.1/5736/AIM-519A.pdf) (PDF) (Technical report). MIT AI Lab. AI Memo 519a. Retrieved 2011-01-07.
21. Leigh Klotz (2007-08-29). "email quoted in "Be Careful What You Joke About" " (http://pogue.blogs.nytimes.com/2007/08/29/be-careful-what-you-joke-about/). *The New York Times*. Retrieved 2010-05-01.
22. "Comment by ZWEI's author Dan Weinreb" (http://steve-yegge.blogspot.be/2008/04/xemacs-is-dead-long-live-xemacs.html#c8187829185600860534). "I wrote the second Emacs ever: the Lisp machine implementation, whose spec was "do what Stallman's PDP-10 (original) Emacs does", and then progressed from there. There's just a whole LOT of it. It took me and Mike McMahon endless hours to implement so many commands to make ZWEI/Zmacs."
23. Owen Theodore Anderson (January 1979). "The Design and Implementation of a Display-Oriented Editor Writing System" (http://dspace.mit.edu/bitstream/handle/1721.1/16038/07332831.pdf?sequence=1) (PDF). Retrieved 2012-09-09.
24. Christopher Kelty; Mario Biagioli; Peter Jaszi; Martha Woodmansee (2015). *Making and Unmaking Intellectual Property*. "*Computer Corporation of America (CCA)* EMACS, written by Steve Zimmerman"
25. "Emacs" (http://www.softwarepreservation.org/projects/emacs). December 17, 2017.
26. "Differences between GNU Emacs and CCA Emacs. Copyyright" (http://www.retro11.de/ouxr/43bsd/usr/src/new/emacs/etc/CCADIFF).
27. "A History of Emacs" (http://www.xemacs.org/Documentation/21.5/html/internals_3.html).
28. "emacs-devel msg 00872 (2014-09-29)" (http://lists.gnu.org/archive/html/emacs-devel/2014-09/msg00872.html). "In retrospect 24.3 should have been named 25.1 and 24.4 should have been named 26.1. The *.N* thingy should really be kept only for bug-fix releases and neither of 24.3, 24.4, nor the previously planned 24.5 are bug-fix releases."
29. Stoll, Clifford (1988). "Stalking the wily hacker". *Communications of the ACM*. **31** (5): 484–497. doi:10.1145/42411.42412 (https://doi.org/10.1145%2F42411.42412).

30. "GNU Emacs" (https://www.openhub.net/p/emacs/analyses/latest/languages_summary). Open Hub. Retrieved 2017-11-25.
31. Raymond, Eric (2014-11-12). "New Git repository is up" (http://lists.gnu.org/archive/html/emacs-devel/2014-11/msg00681.html). Retrieved 2017-11-25.
32. Stallman, Richard (2008-02-22). "Re: Looking for a new Emacs maintainer or team" (http://lists.gnu.org/archive/html/emacs-devel/2008-02/msg02140.html). gnu.org Mailing List. Retrieved 2017-11-25.
33. McNamara, Paul (2008-02-25). "Stallman on handing over GNU Emacs, its future and the importance of nomenclature" (http://www.networkworld.com/community/node/25360). *Network World*. Retrieved 2017-11-25.
34. Chirgwin, Richard (2015-11-05). "Emacs gets new maintainer as Richard Stallman signs off" (https://www.theregister.co.uk/2015/11/05/wiegley_new_emacs_maintainer/). *The Register*. Retrieved 2017-11-25.
35. Raymond, Eric (2014-03-29). "Ugliest…repository…conversion…ever" (http://esr.ibiblio.org/?p=5634). Retrieved 2017-11-25. "twenty-nine years of continuous development by no fewer than 579 people"
36. Stephen J., Turnbull. "XEmacs vs. GNU Emacs" (http://www.xemacs.org/About/XEmacsVsGNUemacs.html). Retrieved 2012-10-02.
37. "XEmacs is Dead. Long Live XEmacs!" (http://steve-yegge.blogspot.com/2008/04/xemacs-is-dead-long-live-xemacs.html).
38. FrontPage - Meadow Wiki (http://www.meadowy.org/meadow/pukiwiki-en/) Archived (https://web.archive.org/web/20120216081734/http://www.meadowy.org/meadow/pukiwiki-en/) 2012-02-16 at the Wayback Machine
39. "SXEmacs Website" (http://www.sxemacs.org). Sxemacs.org. 2009-10-11. Retrieved 2009-11-08.
40. "My Lisp Experiences and the Development of GNU Emacs" (https://www.gnu.org/gnu/rms-lisp.html). "There were people in those days, in 1985, who had one-megabyte machines without virtual memory. They wanted to be able to use GNU Emacs. This meant I had to keep the program as small as possible."
41. "GNU Zile (Zile is Lossy Emacs) - Summary" (https://savannah.gnu.org/projects/zile/). "Modern computers have more than enough resources to start and run a full Emacs in a fraction of a second, and Emacs is probably what you want. Zile is a small, fast, and powerful Emacs clone. It is useful for small footprint installations (e.g. on floppy disk), machines with little memory, or quick editing sessions, especially on remote machines or as a different user, e.g. root."
42. "Zile home page" (https://www.gnu.org/software/zile/).
43. "Lugaru Software Homepage" (http://www.lugaru.com/).
44. "AMACS source" (http://www.mmnt.net/db/0/0/moscoso.org/pub/emulators/apple2/8bit/source).
45. "apple2.org" (http://apple2.org.za/gswv/a2zine/GS.WorldView/v1999/Feb/A2.NET.GOODIES.2.99/A2Programmer%27s.Languages.txt).
46. Paul S. Dayan (1992). *The OS-9 Guru - 1 : The Facts*. Galactic Industrial Limited. ISBN 0-9519228-0-7.
47. "Cocoa text system" (https://www.hcs.harvard.edu/~jrus/site/cocoa-text.html).
48. "Setting up keybindings for the Command Window and Editor" (https://blogs.mathworks.com/community/2007/05/11/setting-up-keybindings-for-the-command-window-and-editor/). *Mathworks Blogs*. 2007-05-11. Retrieved 2019-08-18.
49. "Init file" (https://www.gnu.org/software/emacs/manual/html_node/emacs/Init-File.html).
50. "Saint IGNUcius - Richard Stallman" (http://stallman.org/saint.html). Retrieved 29 January 2015.
51. "The unabridged selective transcript of Richard M Stallman's talk at the ANU" (http://linuxhelp.blogspot.com/2006/04/unabridged-selective-transcript-of.html). Retrieved 29 January 2015.

52. alt.religion.emacs newsgroup (news:alt.religion.emacs)
53. "Saint IGNUcius - Richard Stallman" (http://www.stallman.org/saint.html). *www.stallman.org*. Retrieved 27 March 2018.
54. "How To Avoid The Emacs Pinky Problem" (http://xahlee.org/emacs/emacs_pinky.html). Xahlee.org. Retrieved 2009-11-08.
55. "Emacs: Why You Should Not Swap {Caps Lock, Control} Keys" (http://ergoemacs.org/emacs/swap_CapsLock_Ctrl.html). "In touch typing, modifiers comes in pairs, such as ⇧ Shift key. The accepted ergonomic way to press modifier combinations is using one hand to press the modifier key and the other to press the letter key. Using 2 hands avoids single-hand straining its muscle."
56. "EmacsWiki: Repeated Strain Injury" (http://www.emacswiki.org/emacs/RepeatedStrainInjury). Retrieved 29 January 2015.
57. "Moving The Ctrl Key" (http://www.emacswiki.org/emacs/MovingTheCtrlKey). EmacsWiki. Retrieved 2009-11-08.
58. "ErgoEmacs Homepage" (https://ergoemacs.github.io/). Retrieved 2018-09-01.
59. "Keymacs layout" (https://github.com/keyboard-ergonomics/keymacs).
60. "EmacsWiki: Ace Jump" (http://www.emacswiki.org/emacs/AceJump). Retrieved 29 January 2015.
61. "Spacemacs" (https://github.com/syl20bnr/spacemacs/). Retrieved 2015-04-20.
62. BayleShanks (2009-10-07). "Sticky Modifiers" (http://www.emacswiki.org/emacs/StickyModifiers). EmacsWiki. Retrieved 2009-11-08.
63. "Viper Mode" (http://www.emacswiki.org/emacs/ViperMode). EmacsWiki. Retrieved 2009-11-08.
64. "At Home Modifier by Evdev" (https://gitlab.com/at-home-modifier/at-home-modifier-evdev/wikis/home). Retrieved 2015-04-14.
65. "Why Emacs's Keyboard Shortcuts are Painful" (http://ergoemacs.org/emacs/emacs_kb_shortcuts_pain.html).
66. "VAXen" (http://www.catb.org/~esr/jargon/html/V/VAXen.html). Catb.org. Retrieved 2009-11-08.

Bibliography

- Ciccarelli, Eugene (1978). *An Introduction to the Emacs Editor*. Cambridge, Massachusetts: MIT Artificial Intelligence Laboratory. AIM-447. PDF (https://archive.org/details/bitsavers_mitaiaimAI_4967481)
- Stallman, Richard M. (1981) [1979]. *EMACS: The Extensible, Customizable, Self-Documenting Display Editor*. Cambridge Massachusetts: MIT Artificial Intelligence Laboratory. AIM-519A. PDF (https://archive.org/details/bitsavers_mitaiaimAI_2207521) HTML (https://www.gnu.org/software/emacs/emacs-paper.html)
- Stallman, Richard M. (2002). *GNU Emacs Manual* (https://www.gnu.org/software/emacs/manual/) (15th ed.). GNU Press. ISBN 1-882114-85-X.
- Stallman, Richard M. (2002). "My Lisp Experiences and the Development of GNU Emacs" (https://www.gnu.org/gnu/rms-lisp.html). Retrieved 2007-02-01.
- Chassel, Robert J. (2004). *An Introduction to Programming in Emacs Lisp* (https://www.gnu.org/software/emacs/emacs-lisp-intro/). GNU Press. ISBN 1-882114-56-6.
- Glickstein, Bob (April 1997). *Writing GNU Emacs Extensions* (https://archive.org/details/writinggnuemacse00glic). O'Reilly & Associates. ISBN 1-56592-261-1.
- Cameron, Debra; Elliott, James; Loy, Marc; Raymond, Eric; Rosenblatt, Bill (December 2004). *Learning GNU Emacs, 3rd Edition* (http://www.oreilly.com/catalog/gnu3/). O'Reilly & Associates. ISBN 0-596-00648-9.

- Finseth, Craig A. (1991). *The Craft of Text Editing -or- Emacs for the Modern World* (http://www.finseth.com/craft/). Springer-Verlag & Co. ISBN 978-1-4116-8297-9.
- Thompson, Adrienne G. (2009). "MACSimizing TECO" (http://www.codeartnow.com/hacker-art-1/macsimizing-teco). Retrieved 2012-02-26.

External links

- Official website (https://www.gnu.org/software/emacs)
- Reviewed entry (http://directory.fsf.org/wiki/Emacs) in the Free Software Directory.
- Wikemacs (http://www.wikemacs.org/index.php)
- EmacsWiki (http://www.emacswiki.org/)
- EmacsFamily (http://texteditors.org/cgi-bin/wiki.pl?EmacsFamily)
- List of Emacs implementations (http://www.finseth.com/emacs.html)
- Architectural overview (http://chrismennie.ca/EMACS-Conceptual-Architecture.pdf)
- One of the Oldest Rivalries in Computing: Emacs vs Vi (http://www.slate.com/articles/technology/bitwise/2014/05/oldest_software_rivalry_emacs_and_vi_two_text_editors_used_by_programmers.single.html)