

Emacs: The Editor For The Next 40 Years

Jeetaditya Chatterjee

January 12, 2022

What is emacs

Emacs at its core is not a text editor but a lisp vm that specialises in doing textual work, the current incarnation is (gnu emacs) 36 years old, almost seeing constant development in that time frame. this has lead it to become one of the most powerful and unique editors of our time and one that is extremely addicting to use.

The Humble beginnings

the first emacs was a set of macros on top of the TECO editor, this was not macro's in the keybind or vim sense but more a kind of program to tell the editor to do, like emacs TECO was an interpreter for its own language, it was very powerful for its time.

tho let it be known this was not an editor like we have today, it was a line editor which means you literally had to type commands to move up and down in the line, a command to insert lines and if you messed up, you started again.

After that there were some clones (most notably one by James Gosling (the creator of Java)) but none of them would last `teco_EP.gif`

How has emacs has survived this long?

In other words why has it not gone the way of netbeans after more modern editors came to the table?

Cults help

Emacs has an almost cult like following this is for good reason as emacs is excellent for editing text, people start off wanting to program, then miss the those tools when writing up an email or a blog post, suddenly they want those keybinds when surfing the web and the spiral out from there

There is a running joke in the emacs community about everything going into emacs, the problem is that some people heard that joke and was like “yeah that would be nice.” things like,

a version of Emacs has been around since the 1976, most people in this room were either not born or playing with rattles or Atari's In all that time its only grown in power, spear heading a lot of early concepts that then get reinvented later to become mainstream (usually becoming less powerful in the process). things like jumping to errors, early ide / shell interfaces, in editor debugger interfaces and so on all have a history of being written in emacs lisp first.

also for the people who used latex editors in the early 90's a lot of those features came from emacs's latex-mode and later a plugin called auxtex (a plugin I use to this day and can say is still kinda best in class)

- emacs terminals
- emacs shells (as in a shell implemented in emacs lisp)
- emacs window managers
- emacs web browsers (as in they embed a full web browser)
- emacs init systems...

also in some places its still more advanced than most editors

Things like an infinite clipboard which instead of clobbering the last item its pushed onto a stack, which can then be traversed.

A tree based undo system which as well can be traversed and allows you to lean into many time travel metaphors

and if you want to get into certain languages such as common lisp, Clojure and racket the emacs experience is very hard to beat.

Hackability

emacs has a very low barrier of entry to get working on it, unlike a lot of editors today the smallest unit of extension is not a package or project, but a function in most cases though most of the time you can configure it with small tweaks, setting variables and whatnot.

Emacs is not structured like editors today, as in there is little structure too it at all, even packages are more a collection of forms using a convention rather than some strict hierarchy of statements talking through an api.

this means that pretty much everything is up for grabs (except maybe some c special forms). This leads into the next couple of points where most parts of the editor are written emacs lisp and every function / special form can be “advised” at runtime. its similar to python decorators in the way they manipulate a function in some certain way but with the key difference is that they don’t need to be called where the function is defined. this means a user can edit any function to act in the exact way they want and it will be run like the user wants every time its called, this is extremely powerful.

- It has a low barrier to extension
- Most if not all things can be extended / modified as most things are implemented in emacs lisp
- you can advise functions to do exactly what you want
- and so much more. . .

Its extremely introspect able

every function, variable and special form is documented and easily accessible and checked, either by pressing a key-bind on the form or by searching for it in a help lookup. this again makes it way to easy to hack on it as there are entire weeks I never need to look something up. add to that a very strong documentation culture and I can really look into the guts of how my editor works and actually fiddle with it.

all of this leads to having the power to iterate extremely quickly on a feature drilling right on down to the exact way your editor will work.

in line re evaluation.

Switch to editor

the ecosystem

all of this allows for an extremely interesting ecosystem to pop up, as its easy to iterate on designs, interesting UI concepts can and do pop up all over the place, some quick examples include

org mode

show off the agenda and the ability to generate other files, it makes my life easier as its so maliable

the git interface

its a like an interactive version of the cli.

so

many

more

Why it will outlast current editors

Its pragmatic but not stagnant

Emacs is slower to pick up trends but this is not really a problem as the community can implement it

when it is implemented its usually vetted and documented very well and in a way other people can build off it. this makes it much easier to hack on it.

As time marches on it also plans to add lsp support (the thing that makes vscodes completion go brrr) and tree sitter support (both of which have been implemented out side of emacs core and work really well).

Its very backwards compatible!

When a new feature is added it usually sticks around for a very long time. this means that configs need minimal tweaking between updates or none over years. this leads to people using the same config for decades

Its got a community with no where else to go...

Emacs is pretty much unique in this space, no where else can you find a lisp machine that self documents as you edit it on the fly. its old and wise and with a community of addicts that will happily maintain and update it as time goes on.

Conclusion

emacs is cool

Thanks