Emacs: The Editor of a Lifetime

or... Thirty Years of M-x info

Perry E. Metzger
perry@piermont.com

University of Pennsylvania
Department of Computer and Information Science

NYC Emacs Meetup, August 11, 2014

I learned Emacs in September, 1983

I learned Emacs in September, 1983

It is now August, 2014

I learned Emacs in September, 1983

It is now August, 2014

...in one month, it will have been thirty one years

I learned Emacs in September, 1983

It is now August, 2014

...in one month, it will have been thirty one years

Why would anyone use a program that long?

I learned Emacs in September, 1983

It is now August, 2014

...in one month, it will have been thirty one years

Why would anyone use a program that long?

Surely the state of the art has improved?

Best terminal at Columbia in 1983.

Best terminal at Columbia in 1983.

People used to camp out to use one. Now it is a museum piece.

The OS I used was TOPS-20, on a DECSYSTEM-20

The OS I used was TOPS-20, on a DECSYSTEM-20

I don't miss it one bit.

Crazy nostalgia aside, no one wants to use museum hardware.

Crazy nostalgia aside, no one wants to use museum hardware.

Some of you in this room weren't even *born* in 1983.

Crazy nostalgia aside, no one wants to use museum hardware.

Some of you in this room weren't even *born* in 1983.

So why the hell would I still use Emacs?

In fact, I don't just use Emacs, I live in Emacs.

In fact, I don't just use Emacs, I live in Emacs.
I only recently (temporarily) stopped reading Email in Emacs but I intend to fix that.

In fact, I don't just use Emacs, I live in Emacs.
I only recently (temporarily) stopped reading Email in Emacs but I intend to fix that.
I keep my calendar and phonebook in Emacs (though I want better integration with OS tools).

In fact, I don't just use Emacs, I live in Emacs.
I only recently (temporarily) stopped reading Email in Emacs but I intend to fix that.
I keep my calendar and phonebook in Emacs (though I want better integration with OS tools).
I do all my development work in Emacs.

In fact, I don't just use Emacs, I live in Emacs.
I only recently (temporarily) stopped reading Email in Emacs but I intend to fix that.
I keep my calendar and phonebook in Emacs (though I want better integration with OS tools).
I do all my development work in Emacs.
I (sometimes) use IRC/ICB/etc. in Emacs.

In fact, I don't just use Emacs, I live in Emacs.
I only recently (temporarily) stopped reading Email in Emacs but I intend to fix that.
I keep my calendar and phonebook in Emacs (though I want better integration with OS tools).
I do all my development work in Emacs.
I (sometimes) use IRC/ICB/etc. in Emacs.
I wrote this presentation in Emacs (it's LaTeX).

In fact, I don't just use Emacs, I live in Emacs.
I only recently (temporarily) stopped reading Email in Emacs but I intend to fix that.
I keep my calendar and phonebook in Emacs (though I want better integration with OS tools).
I do all my development work in Emacs.
I (sometimes) use IRC/ICB/etc. in Emacs.
I wrote this presentation in Emacs (it's LaTeX).
I wish I could browse the web and read papers in Emacs, maybe that can be fixed too.

I am not the only one, either.

Another PhD Student at U. Penn said to me:

> *I don't care about the OS I use, it is just a boot loader for Emacs anyway.*

If you take CIS-500 at Penn, you'll watch the professor run Coq from within Proof General Mode in Emacs.

My “normal” non-programmer friend Ian uses Emacs for his calendar, email, to-do lists, practically everything.

Heck, I'm giving this talk to the *Emacs Meetup.*

Heck, I'm giving this talk to the *Emacs Meetup*. I gave the original version to the *Emacs Club* at Penn!

Heck, I'm giving this talk to the *Emacs Meetup.* I gave the original version to the *Emacs Club* at Penn!

Sure, there's a lot of weird people in any CS department, but there's an *Emacs Club.*

Heck, I'm giving this talk to the *Emacs Meetup.*
I gave the original version to the *Emacs Club* at Penn!

Sure, there's a lot of weird people in any CS department, but there's an *Emacs Club.*
There's an *Emacs Meetup* here in a city not lacking for forms of entertainment.

Heck, I'm giving this talk to the *Emacs Meetup.*
I gave the original version to the *Emacs Club* at Penn!

Sure, there's a lot of weird people in any CS department, but there's an *Emacs Club.*
There's an *Emacs Meetup* here in a city not lacking for forms of entertainment.

That implies *fanaticism.*

WHY?

Is it because Emacs is so easy to learn?

Is it because Emacs is so easy to learn?

Those of you who use it every day know the learning curve is (ridiculously) steep.

Is it because Emacs is so pretty?

Is it because Emacs is so pretty?

HELL no.

Is Emacs well promoted by companies selling it?

Is Emacs well promoted by companies selling it?

Don't laugh too hard.

Is Emacs well promoted by companies selling it?

Don't laugh too hard.

UniPress actually tried.

On the other hand, one of my doctoral brothers, in a meeting recently, opined that he didn't see why anyone would bother.

On the other hand, one of my doctoral brothers, in a meeting recently, opined that he didn't see why anyone would bother.

He uses an IDE.

On the other hand, one of my doctoral brothers, in a meeting recently, opined that he didn't see why anyone would bother.

He uses an IDE.

...and I don't think he's ever *really* tried Emacs.

So, **WHY?**

So, **WHY?**

Is it just mental illness?

No.

No.

It isn't *just* mental illness.

First, the uninteresting explanation.

First, the uninteresting explanation.

Natural Selection

In the beginning was the Teletype

Teletypes mean line editors.

Teletypes mean line editors.

Late 1960s: Glass TTYs appear. They save paper.

Teletypes mean line editors.

Late 1960s: Glass TTYs appear. They save paper.

By Mid-1970s: Commercial terminals with positional cursor control.

(1975: VT-52)

Status quo: line editors.

Status quo: line editors.

ca. 1972-1975: Modern terminals

Status quo: line editors.

ca. 1972-1975: Modern terminals

1976: Emacs (Richard Stallman, Guy Steele)

1976: vi (Bill Joy)

Status quo: line editors.

ca. 1972-1975: Modern terminals

1976: Emacs (Richard Stallman, Guy Steele)

1976: vi (Bill Joy)

Coincidence?

At least hundreds of screen editors created were created in the mid-to-late 1970s.

At least hundreds of screen editors created were created in the mid-to-late 1970s.

Emacs and vi *are the sole survivors.*

At least hundreds of screen editors created were created in the mid-to-late 1970s.

Emacs and vi *are the sole survivors.*

We expect they survived *because they were better.*

At least hundreds of screen editors created were created in the mid-to-late 1970s.

Emacs and vi *are the sole survivors.*

We expect they survived *because they were better.*

Were the designers lucky or smart?

At least hundreds of screen editors created were created in the mid-to-late 1970s.

Emacs and vi *are the sole survivors.*

We expect they survived *because they were better.*

Were the designers lucky or smart?

Can we know? Does it matter?

I would argue that the Emacs designers were smart.

I would argue that the Emacs designers were smart.

...not the first editor they dealt with (E, TECO Ctrl-R mode)

I would argue that the Emacs designers were smart.

...not the first editor they dealt with (E, TECO Ctrl-R mode)

...started as "improved" TECO + macro packages.

I would argue that the Emacs designers were smart.

...not the first editor they dealt with (E, TECO Ctrl-R mode)

...started as "improved" TECO + macro packages.

...macros evolved fast in community.

I would argue that the Emacs designers were smart.

...not the first editor they dealt with (E, TECO Ctrl-R mode)

...started as "improved" TECO + macro packages.

...macros evolved fast in community.

...community had great hackers with great ideas.

I would argue that the Emacs designers were smart.

...not the first editor they dealt with (E, TECO Ctrl-R mode)

...started as "improved" TECO + macro packages.

...macros evolved fast in community.

...community had great hackers with great ideas.

but in some sense, it makes no difference.

More interesting explanation.

More interesting explanation.

It works really well!

Great at Editing!

Menus and mice are just too slow.

Great at Editing!

Menus and mice are just too slow.

In Emacs, you change text as fast as you think.

Great at Editing!

Menus and mice are just too slow.

In Emacs, you change text as fast as you think.

Keyboard orientation is feature, not a bug.

Great at Editing!

Menus and mice are just too slow.

In Emacs, you change text as fast as you think.

Keyboard orientation is feature, not a bug.

Hands never leaving keyboard means *speed*!

Great at Editing!

Open and close files, switch buffers, split screen, change window, transpose letters, transpose words, capitalize, downcase, move text up and down, reflow, reformat, i-search, spellcheck, all without hands moving from keyboard.

Great at Editing!

Open and close files, switch buffers, split screen, change window, transpose letters, transpose words, capitalize, downcase, move text up and down, reflow, reformat, i-search, spellcheck, all without hands moving from keyboard.

Programmer features: automatic language indentation, compile code, fix errors, refactor, etc, all without hands moving from keyboard.

Great at Editing!

Open and close files, switch buffers, split screen, change window, transpose letters, transpose words, capitalize, downcase, move text up and down, reflow, reformat, i-search, spellcheck, all without hands moving from keyboard.

Programmer features: automatic language indentation, compile code, fix errors, refactor, etc, all without hands moving from keyboard.

Macros! Do mass edits without programming!

Work flows fast!

It is hard to explain how fast thoughts turn to changed text.

Work flows fast!

It is hard to explain how fast thoughts turn to changed text.

Editing in GUIs drags on, it's *slow*.

Work flows fast!

It is hard to explain how fast thoughts turn to changed text.

Editing in GUIs drags on, it's *slow*.

Many of you understand.

Extensible!

TECO was first (terrible) extension language.

Extensible!

TECO was first (terrible) extension language.

Then Lisp became customary
(Multics Emacs, EINE, Zwei, Gnu Emacs...)

Extensible!

TECO was first (terrible) extension language.

Then Lisp became customary
(Multics Emacs, EINE, Zwei, Gnu Emacs...)

Extensible means EXTENSIONS...

Extensible!

TECO was first (terrible) extension language.

Then Lisp became customary
(Multics Emacs, EINE, Zwei, Gnu Emacs...)

Extensible means EXTENSIONS...
...read email in Emacs

Extensible!

TECO was first (terrible) extension language.

Then Lisp became customary
(Multics Emacs, EINE, Zwei, Gnu Emacs...)

Extensible means EXTENSIONS...
...read email in Emacs
...run debugger in Emacs

Extensible!

TECO was first (terrible) extension language.

Then Lisp became customary
(Multics Emacs, EINE, Zwei, Gnu Emacs...)

Extensible means EXTENSIONS...
...read email in Emacs
...run debugger in Emacs
...IDEs, refactoring tools in Emacs

Extensible!

TECO was first (terrible) extension language.

Then Lisp became customary
(Multics Emacs, EINE, Zwei, Gnu Emacs...)

Extensible means EXTENSIONS...
...read email in Emacs
...run debugger in Emacs
...IDEs, refactoring tools in Emacs
...and of course yet more powerful editing

Extensible!

TECO was first (terrible) extension language.

Then Lisp became customary
(Multics Emacs, EINE, Zwei, Gnu Emacs...)

Extensible means EXTENSIONS...
...read email in Emacs
...run debugger in Emacs
...IDEs, refactoring tools in Emacs
...and of course yet more powerful editing
...etc, etc.

Emacs is a UI for text hacking!

Extensions are applications

Emacs is a UI for text hacking!

Extensions are applications
...but why write apps in (mediocre) Emacs Lisp?

Emacs is a UI for text hacking!

Extensions are applications
...but why write apps in (mediocre) Emacs Lisp?

Extensions seem weird (to non-Emacsers).
Why read or write email inside Emacs?

Emacs is a UI for text hacking!

Extensions are applications
...but why write apps in (mediocre) Emacs Lisp?

Extensions seem weird (to non-Emacsers).
Why read or write email inside Emacs?

Because it's totally natural.
When you're writing email, you're *editing text!*
Why edit with something less powerful?

“I only run one OS at a time.”

“I only run one OS at a time.”

A common complaint: “Emacs is an OS.”

“I only run one OS at a time.”

A common complaint: “Emacs is an OS.”

My reply: “You say that like it’s a bad thing.”

“I only run one OS at a time.”

A common complaint: “Emacs is an OS.”

My reply: “You say that like it’s a bad thing.”

(Though it is more of a UI framework than a true OS.)

The Old NetBSD Boot Loader.

The Old NetBSD Boot Loader.

```
static const char * const names[] = {
    "netbsd", "netbsd.gz",
    "netbsd.old", "netbsd.old.gz",
    "onetbsd", "onetbsd.gz",
#ifdef notyet
    "netbsd.el", "netbsd.el.gz",
#endif /*notyet*/
};
```

Emacs is a UI for text hacking!

Why wouldn't you want to cut and paste from other files with the keyboard when composing mail?

Emacs is a UI for text hacking!

Why wouldn't you want to cut and paste from other files with the keyboard when composing mail?

Why wouldn't you want to edit and debug in the same application?
(Yes, IDEs were created later to let you do this, but they're less powerful.)

Emacs is a UI for text hacking!

Why wouldn't you want to cut and paste from other files with the keyboard when composing mail?

Why wouldn't you want to edit and debug in the same application?
(Yes, IDEs were created later to let you do this, but they're less powerful.)

Why wouldn't you want to *reprogram* and *extend* your UI?

Emacs is a UI for text hacking!

Why wouldn't you want to cut and paste from other files with the keyboard when composing mail?

Why wouldn't you want to edit and debug in the same application?
(Yes, IDEs were created later to let you do this, but they're less powerful.)

Why wouldn't you want to *reprogram* and *extend* your UI?
You are a computer scientist!

Good Computer Professionals Make Tools!

Good Computer Professionals Make Tools!

Machinists make and modify their own tools.

Good Computer Professionals Make Tools!

Machinists make and modify their own tools.

Good computer professionals do the same thing.

Good Computer Professionals Make Tools!

Machinists make and modify their own tools.

Good computer professionals do the same thing.

We know how.

Good Computer Professionals Make Tools!

Machinists make and modify their own tools.

Good computer professionals do the same thing.

We know how.
It makes our lives vastly better.

Good Computer Professionals Make Tools!

Machinists make and modify their own tools.

Good computer professionals do the same thing.

We know how.
It makes our lives vastly better.
Why wouldn't we?

Portable, Open Source

Portable, Open Source

Non-portable programs die with their system.
Closed source programs die with their company.

Portable, Open Source

Non-portable programs die with their system.
Closed source programs die with their company.

Even the first Emacs was Open Source.
Once ported to C & Unix, Emacs could live *forever.*

Portable, Open Source

Non-portable programs die with their system.
Closed source programs die with their company.

Even the first Emacs was Open Source.
Once ported to C & Unix, Emacs could live *forever.*

...Learn Once, Use For Decades!

Professionals Need Efficient, Not Pretty!

Professionals Need Efficient, Not Pretty!

Restaurant kitchens aren't pretty.

Professionals Need Efficient, Not Pretty!

Restaurant kitchens aren't pretty.
The people that work in them need to prepare hundreds of dinners in several hours, not to show off their hand-made cabinets and granite counters to guests.

Professionals Need Efficient, Not Pretty!

Restaurant kitchens aren't pretty.
The people that work in them need to prepare hundreds of dinners in several hours, not to show off their hand-made cabinets and granite counters to guests.

Similarly, CS professionals need tools to get their own work done efficiently. Pretty isn't the point.

Editing is what CS types do.

Editing is what CS types do.

Computer scientists do just two things:

Editing is what CS types do.

Computer scientists do just two things:
...reading text
...and generating it.

Editing is what CS types do.

Computer scientists do just two things:
...reading text
...and generating it.

No one pays us to read if we don't write afterwards.

Editing is what CS types do.

Computer scientists do just two things:
...reading text
...and generating it.

No one pays us to read if we don't write afterwards.

We edit text for a living!

Editing is what CS types do.

Computer scientists do just two things:
...reading text
...and generating it.

No one pays us to read if we don't write afterwards.

We edit text for a living!

Whether software, documentation, email, web pages, we all spend our entire day editing text.

Editing is what CS types do.

Computer scientists do just two things:
...reading text
...and generating it.

No one pays us to read if we don't write afterwards.

We edit text for a living!

Whether software, documentation, email, web pages, we all spend our entire day editing text.

Your productivity depends on how efficiently you can edit text!

Endurance is a feature!

Endurance is a feature!

To be productive, you have to learn your editor well.

Endurance is a feature!

To be productive, you have to learn your editor well.

I've had to learn just *one* editor well!

Endurance is a feature!

To be productive, you have to learn your editor well.

I've had to learn just *one* editor well!

I've been able to use it for over thirty years!
I can probably use it another thirty years!

Endurance is a feature!

To be productive, you have to learn your editor well.

I've had to learn just *one* editor well!

I've been able to use it for over thirty years!
I can probably use it another thirty years!

So it is okay that it has lots of (useful) features.
There's time to make the investment pay!

Endurance = Investment Pays!

Endurance = Investment Pays!

It probably took me a couple of weeks to get good with Emacs.

Endurance = Investment Pays!

It probably took me a couple of weeks to get good with Emacs.
...probably a month or so to get really good.

Endurance = Investment Pays!

It probably took me a couple of weeks to get good with Emacs.
...probably a month or so to get really good.
...but it has probably saved me at least a few minutes, sometimes an hour or more *every single day!*

Endurance = Investment Pays!

It probably took me a couple of weeks to get good with Emacs.
...probably a month or so to get really good.
...but it has probably saved me at least a few minutes, sometimes an hour or more *every single day!*
...for thirty years!

Endurance = Investment Pays!

It probably took me a couple of weeks to get good with Emacs.
...probably a month or so to get really good.
...but it has probably saved me at least a few minutes, sometimes an hour or more *every single day!*
...for thirty years!
...and probably will until I drop dead in another thirty.

Endurance = Investment Pays!

It probably took me a couple of weeks to get good with Emacs.
...probably a month or so to get really good.
...but it has probably saved me at least a few minutes, sometimes an hour or more *every single day!*
...for thirty years!
...and probably will until I drop dead in another thirty.
Amazing return on investment!

So why has it survived?

So why has it survived?

Great editor...

So why has it survived?

Great editor...
...that you can practically live in

So why has it survived?

Great editor...
...that you can practically live in
...that you can extend and customize

So why has it survived?

Great editor...
...that you can practically live in
...that you can extend and customize
...that you will never have to stop using.

So why has it survived?

Great editor...
...that you can practically live in
...that you can extend and customize
...that you will never have to stop using.

If you're a computer professional, that's irresistible.

What could make it die?

What could make it die?

Failing to adapt!

It *has* adapted...

It *has* adapted...

Emacs has changed in 38 years.

It *has* adapted...

Emacs has changed in 38 years.

Improved extension language.

It *has* adapted...

Emacs has changed in 38 years.

- Improved extension language.
- Window system support

It *has* adapted...

Emacs has changed in 38 years.

- Improved extension language.
- Window system support
- New modes

It *has* adapted...

Emacs has changed in 38 years.

- Improved extension language.
- Window system support
- New modes
- New capabilities (networking)

...but it needs to go further.

...but it needs to go further.

Extension language still mediocre.

...but it needs to go further.

Extension language still mediocre.
Better support for modern refactoring.

...but it needs to go further.

Extension language still mediocre.
Better support for modern refactoring.
Not enough of an OS! (Threading sucks!)

...but it needs to go further.

Extension language still mediocre.
Better support for modern refactoring.
Not enough of an OS! (Threading sucks!)
Not enough of an OS! (I still need to leave it!)

...but it needs to go further.

Extension language still mediocre.
Better support for modern refactoring.
Not enough of an OS! (Threading sucks!)
Not enough of an OS! (I still need to leave it!)
Needs more PIM integration (e.g. protocols).

Suggestions to Future-Proof Emacs

TECO was problematic...

TECO was problematic...

```
GZOJ\UNQN"E 4OUN ’ BUH BUV HK
QN< J BUQ QN*10/3UI
QI< \+2*10+(QQ*QI)UA B L K QI*2-1UJ QA/QJUQ
QA-(QQ*QJ)-2\ 10@I// -1\%I >
QQ/1OUT QH+QT+48UW QW-58"E 48UW \%V ’ QV"N QV
QV^T @^A/
/HKEX$$
```

TECO was problematic...

```
GZOJ\UNQN"E 4OUN ’ BUH BUV HK
QN< J BUQ QN*10/3UI
QI< \+2*10+(QQ*QI)UA B L K QI*2-1UJ QA/QJUQ
QA-(QQ*QJ)-2\ 10@I// -1\%I >
QQ/10UT QH+QT+48UW QW-58"E 48UW \%V ’ QV"N QV
QV^T @^A/
/HKEX$$
```

And so Lisp was a big improvement.

But... ELisp Still Sucks!

But... ELisp Still Sucks!

Some improvements have been made

But... ELisp Still Sucks!

Some improvements have been made
...lexical scope finally!

But... ELisp Still Sucks!

Some improvements have been made
...lexical scope finally!

But

But... ELisp Still Sucks!

Some improvements have been made
...lexical scope finally!

But
...API is awful.

But... ELisp Still Sucks!

Some improvements have been made
...lexical scope finally!

But
...API is awful.
...no modules.

But... ELisp Still Sucks!

Some improvements have been made
...lexical scope finally!

But
...API is awful.
...no modules.
...threading.

But... ELisp Still Sucks!

Some improvements have been made
...lexical scope finally!

But
...API is awful.
...no modules.
...threading.
...not really a good lisp.

But... ELisp Still Sucks!

Some improvements have been made
...lexical scope finally!

But
...API is awful.
...no modules.
...threading.
...not really a good lisp.
...really, really, needs to be replaced.

But... ELisp Still Sucks!

Some improvements have been made
...lexical scope finally!

But
...API is awful.
...no modules.
...threading.
...not really a good lisp.
...really, really, needs to be replaced.

Solution: Build a better lisp in parallel!

Better Refactoring

Better Refactoring

Tools like Eclipse and XCode are symbol aware.

Better Refactoring

Tools like Eclipse and XCode are symbol aware.

Some tools like that for Emacs (CEDIT etc.), but we need much better.

Better Refactoring

Tools like Eclipse and XCode are symbol aware.

Some tools like that for Emacs (CEDIT etc.), but we need much better.

XCode uses LLVM's libclang for C/C++, Emacs could too.

Better Refactoring

Tools like Eclipse and XCode are symbol aware.

Some tools like that for Emacs (CEDIT etc.), but we need much better.

XCode uses LLVM's libclang for C/C++, Emacs could too.

Better extension language would help enormously!

Threading and hangs

Threading and hangs

Emacs blocks all windows when computing in any window!

Threading and hangs

Emacs blocks all windows when computing in any window!

It is an OS, but not enough of one!

Threading and hangs

Emacs blocks all windows when computing in any window!

It is an OS, but not enough of one!

I often run three or more instances to get around this.

Threading and hangs

Emacs blocks all windows when computing in any window!

It is an OS, but not enough of one!

I often run three or more instances to get around this.

Per buffer threads, lock only global data?

My #1 problem!

My #1 problem!

I have to leave Emacs!

My #1 problem!

I have to leave Emacs!

...mostly to browse the web

My #1 problem!

I have to leave Emacs!

...mostly to browse the web

...or to view HTML. (HTML email now frequent.)

Solution: Give in to the OS impulse!

Solution: Give in to the OS impulse!

Embed Webkit in Emacs!

Solution: Give in to the OS impulse!

Embed Webkit in Emacs!
Render pages directly in Emacs windows

Solution: Give in to the OS impulse!

Embed Webkit in Emacs!
Render pages directly in Emacs windows
UI managed by Emacs

Solution: Give in to the OS impulse!

Embed Webkit in Emacs!
Render pages directly in Emacs windows
UI managed by Emacs
Buffer shared by Webkit, Emacs

Solution: Give in to the OS impulse!

Embed Webkit in Emacs!
Render pages directly in Emacs windows
UI managed by Emacs
Buffer shared by Webkit, Emacs
isearch web page, copy region, C-x o, paste into email!

Similarly...

Also, possibly add an SVG renderer (or use Webkit's) and a PDF viewer.

Similarly...

Also, possibly add an SVG renderer (or use Webkit's) and a PDF viewer.

...isearch through PDF, copy region, C-x o, paste!

Embrace being an OS!

With embedded Webkit & PDF, might never need to leave Emacs any more!

Embrace being an OS!

With embedded Webkit & PDF, might never need to leave Emacs any more!

Back to the future!

Integrate More!

I want calendar modes to do CalDAV.

Integrate More!

I want calendar modes to do CalDAV.

I want mail reader modes to really do IMAP.

Integrate More!

I want calendar modes to do CalDAV.

I want mail reader modes to really do IMAP.

Contacts should really do CardDAV

Integrate More!

I want calendar modes to do CalDAV.

I want mail reader modes to really do IMAP.

Contacts should really do CardDAV

Etc.

A Rant About Keyboards, Part 1

A Rant About Keyboards, Part 2

A Rant About Keyboards, Part 3

When will Emacs die?

When will Emacs die?

It will die. Nothing lasts forever.

When will Emacs die?

It will die. Nothing lasts forever.

Keyboards ceasing to be the fastest way to enter text?

When will Emacs die?

It will die. Nothing lasts forever.

Keyboards ceasing to be the fastest way to enter text?

But multiple buffers, programmability still needed...

When will Emacs die?

It will die. Nothing lasts forever.

Keyboards ceasing to be the fastest way to enter text?

But multiple buffers, programmability still needed...

...so maybe that's just a new input mode?

When will Emacs die?

It will die. Nothing lasts forever.

Keyboards ceasing to be the fastest way to enter text?

But multiple buffers, programmability still needed...

...so maybe that's just a new input mode?

So who knows?

Questions?