Vim

The last editor you will need

phil

Slides

If you want to follow along

- Slides: https://x4m3.rocks/talks/vim.pdf

Agenda

- History
- Modal editor
- Navigation
- Text edition
- Configuration

History

Vi

The OG

- 1976, Bill Joy

- First visual editor for UNIX
- Very popular, included by default in UNIX distributions
- Wide availability on many platforms

Vim
Vi IMproved

- 1991, Bram Moolenaar

- Syntax highlighting
- Script language for plugins
- GUI (gvim)
- Many more features!

Neovim

The new kid on the block

- Fork of vim, 2015

- Built in Language Server Protocol (LSP)
- Asynchronous IO
- Scripting with Lua programming language

Modal editor

Modal editor

- Modal interface
- Different actions depending on the current mode

- Total of 12 modes
- 6 Basic
- 6 Variants

stackoverflow Products Search... 596 ●9 ●8

Home

PUBLIC

Questions

Tags

Users

Companies

COLLECTIVES

Explore Collectives

TEAMS

Create free Team

Looking for your Teams?

How do I exit Vim?

Asked 10 years, 5 months ago Modified 9 days ago Viewed 2.8m times

Ask Question

5003

I am stuck and cannot escape. It says:

```
type :quit<Enter> to quit VIM
```

But when I type that it simply appears in the object body.

vim vi

Share Edit Follow Flag

edited Jun 4, 2022 at 11:47
UnrealApex
446 ●7 ●18

asked Aug 6, 2012 at 12:25
jclancy
47.9k ●5 ●28 ●33

The Overflow Blog

- AI applications open new security vulnerabilities
- How chaos engineering preps developers for the ultimate game day (Ep. 531)

Featured on Meta

- 2022 Community-a-thon Recap
- Accessibility Update: Colors
- Introducing a new close reason specifically for non-English questions

Important modes

- Normal
- Visual
- Insert
- Command-Line

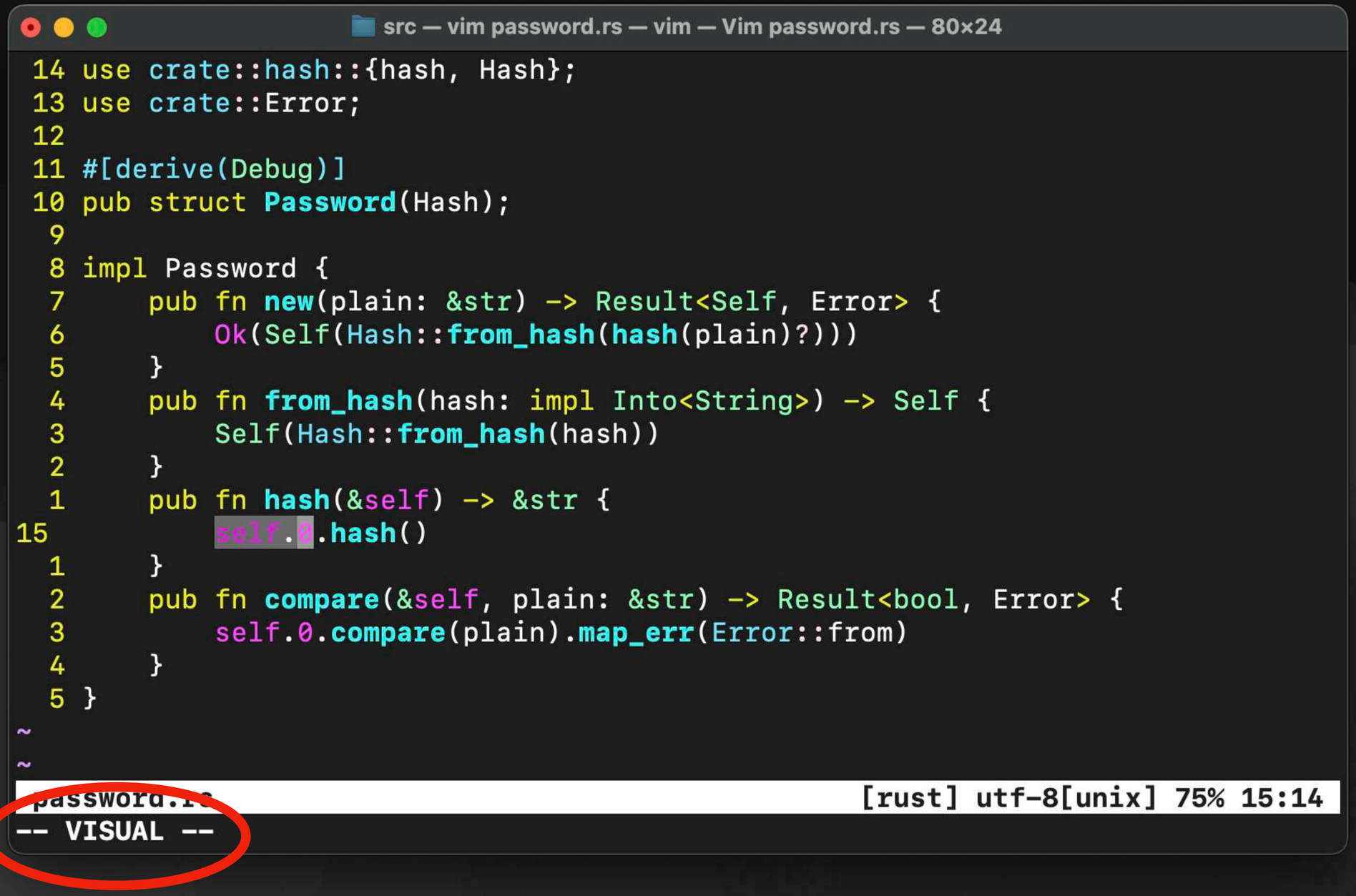
Normal

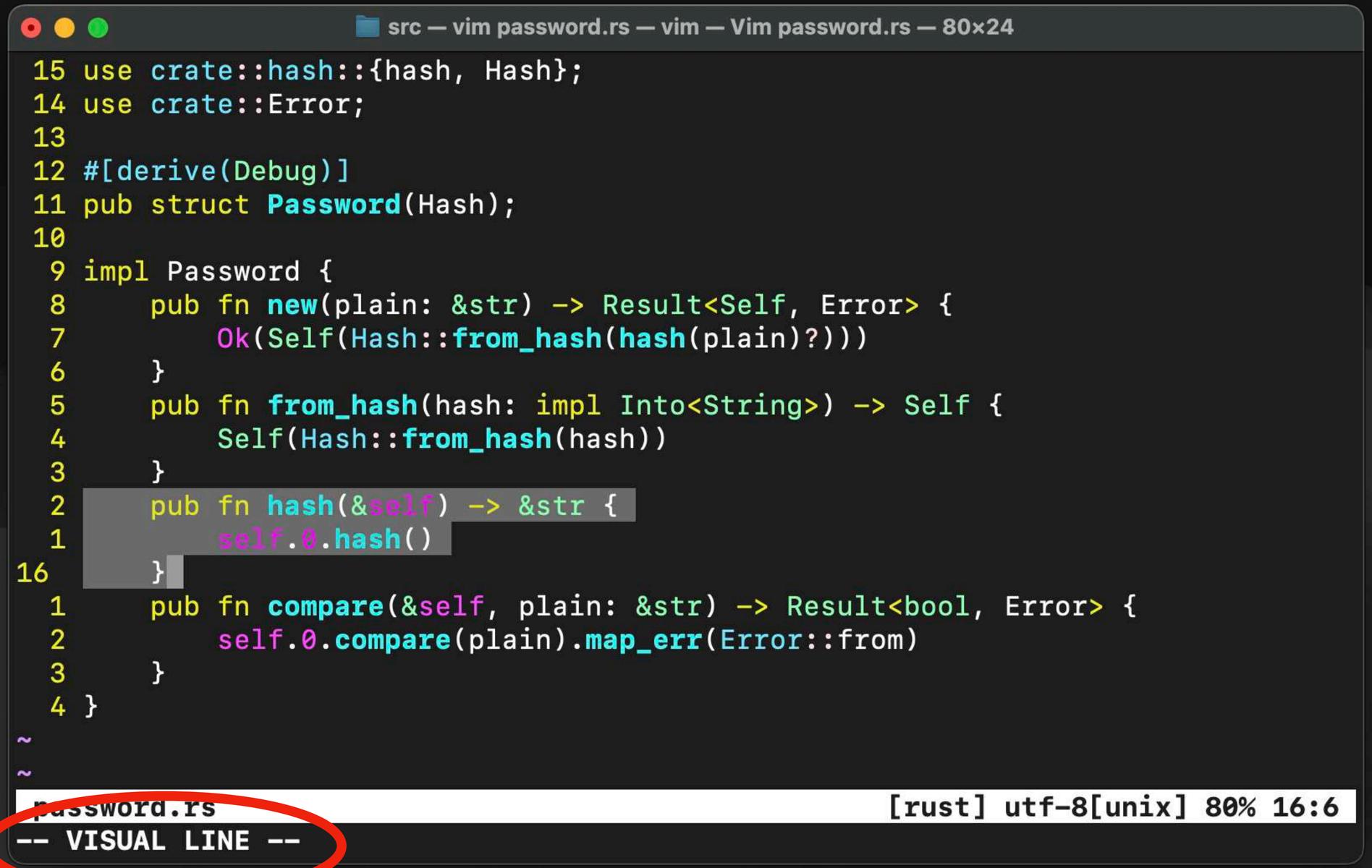
- Default mode
- Used to move around in the file

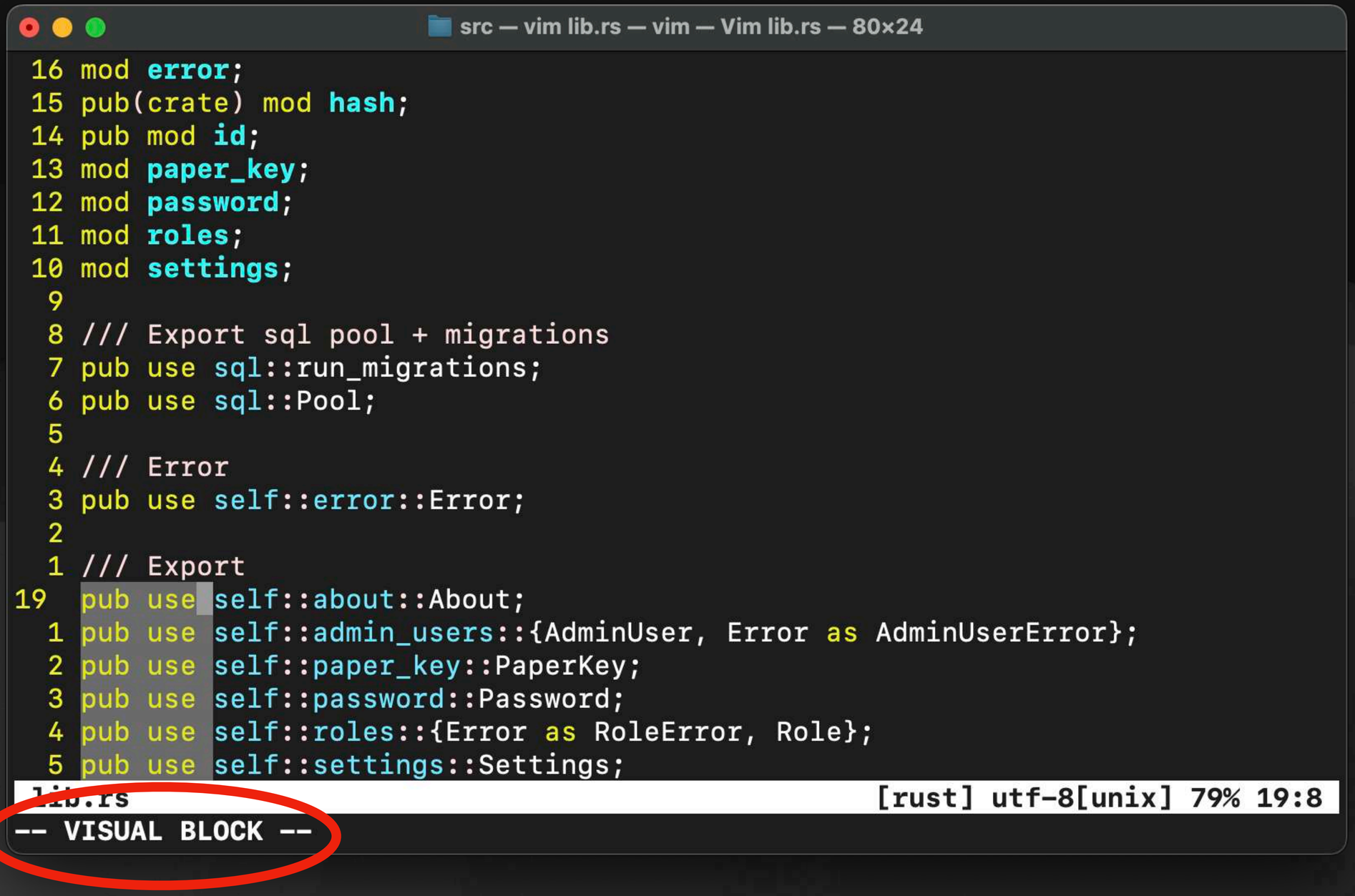
Visual

- Select a piece of text
- Used fo running commands on a selection

- Character mode (v)
- Line mode (V)
- Block mode (ctrl + v)


src — vim password.rs — vim — Vim password.rs — 80×24
14 use crate::hash::{hash, Hash};
13 use crate::Error;
12
11 #[derive(Debug)]
10 pub struct Password(Hash);
9
8 impl Password {
7 pub fn new(plain: &str) -> Result<Self, Error> {
6 Ok(Self(Hash::from_hash(hash(plain)?)))
5 }
4 pub fn from_hash(hash: impl Into<String>) -> Self {
3 Self(Hash::from_hash(hash))
2 }
1 pub fn hash(&self) -> &str {
15 self.0.hash()
1 }
2 pub fn compare(&self, plain: &str) -> Result<bool, Error> {
3 self.0.compare(plain).map_err(Error::from)
4 }
5 }
~
~
[rust] utf-8[unix] 75% 15:14
-- VISUAL --

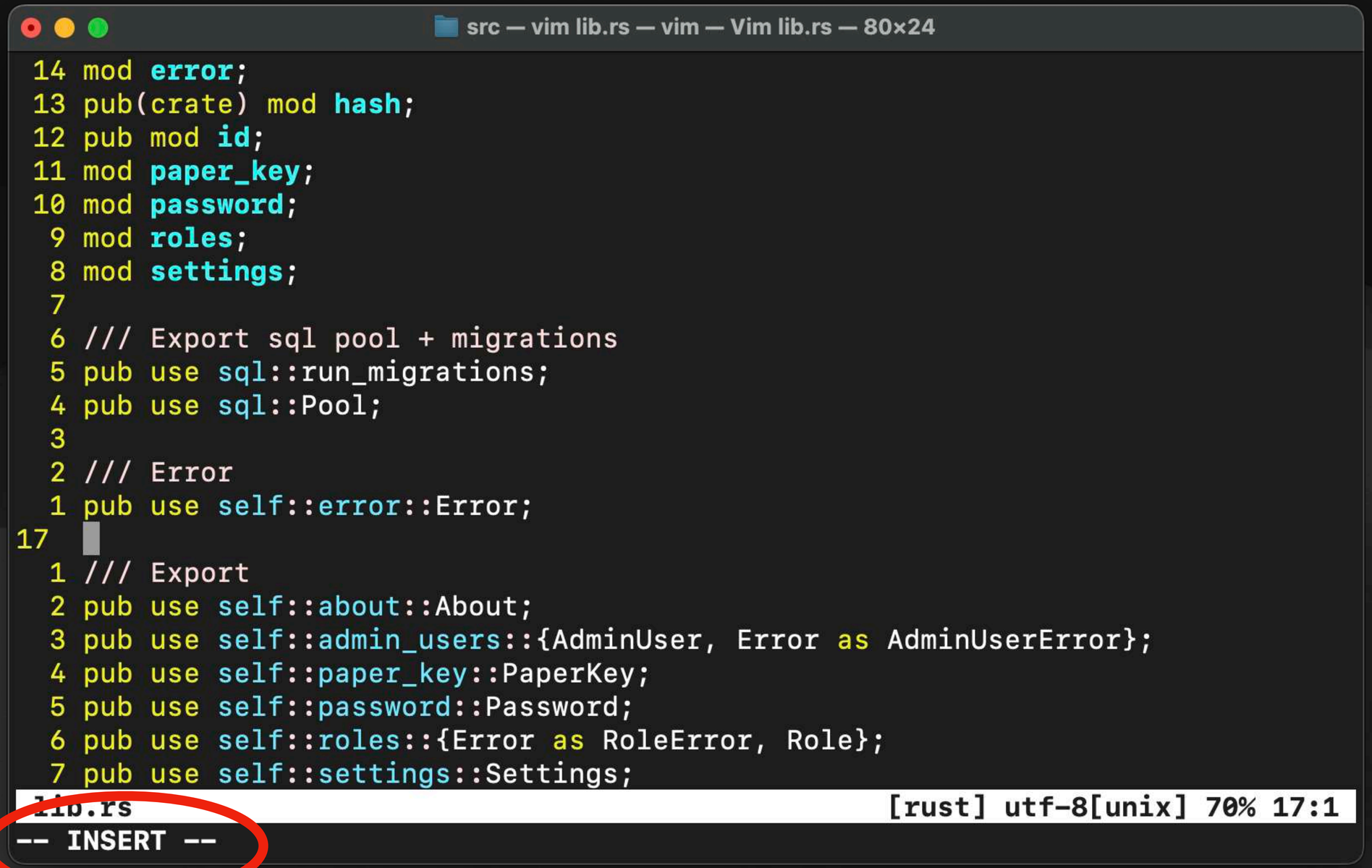

src — vim password.rs — vim — Vim password.rs — 80×24
15 use crate::hash::{hash, Hash};
14 use crate::Error;
13
12 #[derive(Debug)]
11 pub struct Password(Hash);
10
9 impl Password {
8 pub fn new(plain: &str) -> Result<Self, Error> {
7 Ok(Self(Hash::from_hash(hash(plain)?)))
6 }
5 pub fn from_hash(hash: impl Into<String>) -> Self {
4 Self(Hash::from_hash(hash))
3 }
2 pub fn hash(&self) -> &str {
1 self.0.hash()
16 }
1 pub fn compare(&self, plain: &str) -> Result<bool, Error> {
2 self.0.compare(plain).map_err(Error::from)
3 }
4 }
~
~
password.rs [rust] utf-8[unix] 80% 16:6
-- VISUAL LINE --


src — vim lib.rs — vim — Vim lib.rs — 80×24
16 mod error;
15 pub(crate) mod hash;
14 pub mod id;
13 mod paper_key;
12 mod password;
11 mod roles;
10 mod settings;
9
8 /// Export sql pool + migrations
7 pub use sql::run_migrations;
6 pub use sql::Pool;
5
4 /// Error
3 pub use self::error::Error;
2
1 /// Export
19 pub use self::about::About;
1 pub use self::admin_users::{AdminUser, Error as AdminUserError};
2 pub use self::paper_key::PaperKey;
3 pub use self::password::Password;
4 pub use self::roles::{Error as RoleError, Role};
5 pub use self::settings::Settings;
lib.rs [rust] utf-8[unix] 79% 19:8
-- VISUAL BLOCK --

Insert

- Insert some piece of text

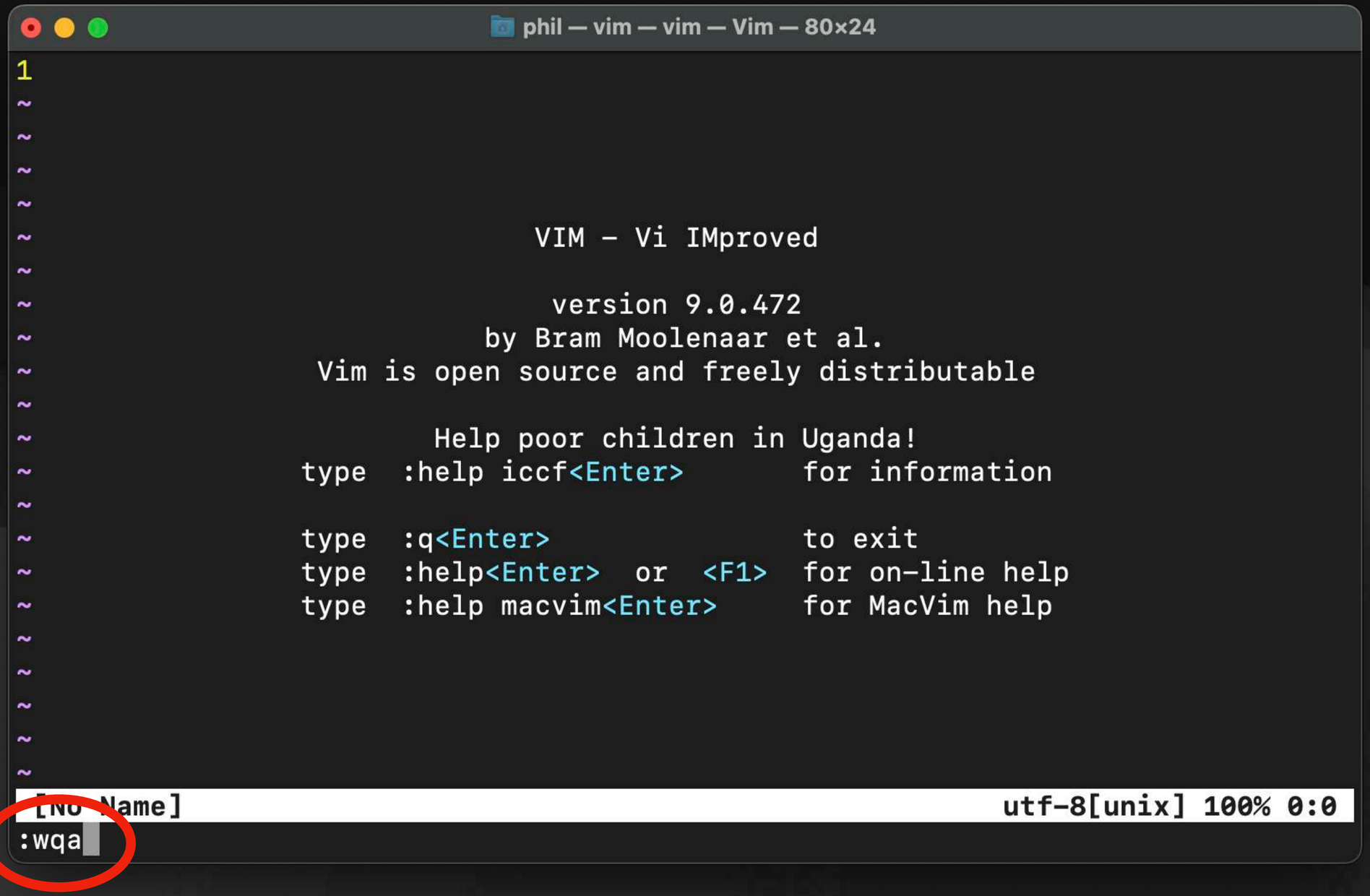
- Insert at current cursor (i)
- Insert at start of line (I)
- Append after current cursor (a)
- Append at end of line (A)


src — vim lib.rs — vim — Vim lib.rs — 80×24
14 mod error;
13 pub(crate) mod hash;
12 pub mod id;
11 mod paper_key;
10 mod password;
9 mod roles;
8 mod settings;
7
6 /// Export sql pool + migrations
5 pub use sql::run_migrations;
4 pub use sql::Pool;
3
2 /// Error
1 pub use self::error::Error;
17
1 /// Export
2 pub use self::about::About;
3 pub use self::admin_users::{AdminUser, Error as AdminUserError};
4 pub use self::paper_key::PaperKey;
5 pub use self::password::Password;
6 pub use self::roles::{Error as RoleError, Role};
7 pub use self::settings::Settings;
lib.rs [rust] utf-8[unix] 70% 17:1
-- INSERT --

Command-Line

- Mode to run quick commands
- Used for editor commands

- Examples: Write, Quit, Open a file, Split window


phil — vim — vim — Vim — 80×24
1
VIM - Vi IMproved
version 9.0.472
by Bram Moolenaar et al.
Vim is open source and freely distributable
Help poor children in Uganda!
type :help iccf<Enter> for information
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help macvim<Enter> for MacVim help
[No Name] utf-8[unix] 100% 0:0
:wqa

Navigation

Move in the file

- Arrow keys
- HJKL keys

- Learn to use HJKL
- Keep your fingers on the home row

HJKL

- Arrow keys on ADM-3A keyboard
- Bill Joy created vi on this computer

- H: Left
- J: Down
- K: Up
- L: Right


ESC
HOME
H
J
K
L
LINE FEED
RETURN
HERE IS
CTRL
RUB
BREAK
SHIFT
SHIFT
REPEAT
CLEAR

Chain navigation

- Combine navigation with a number to repeat the navigation
- Example: 12j: Move 12 times down

Additional movements

On the same line

- 0 (zero): Move to beginning of line
- $ (dollar): Move to end of line

- w: Next word
- b: Beginning of word
- B: Previous word
- e: End of word

Additional movements

Across multiple lines

- :123 (column 123): Go to line 123
- gg: Go to first line of file
- G: Go to last line of file
- %: Go to matching bracket. Works on { } [ ] ( )

Search

- /text: Search “text” in file
- n: Move to next search result
- N: Move to previous search result
- :noh (column noh): Turn off highlighting until next search

Text edition

Delete / Cut

- dd: Delete / Cut the current line

- Chain commands: Example: d2j to delete 2 lines down
- Select text in visual mode and use command `d`

Copy (yank)

- yy: Copy the current line

- Select text in visual mode and use command `y`

Paste

- p: Paste clipboard after cursor
- P: Paste clipboard before cursor

- By default, Vim uses its own clipboard system

Undo / Redo

- u: Undo last action
- ctrl + r: Redo

Replace
In whole file

- :s/find/replace/g (column s / find / replace / g)
- Find “text”, substitute it by “replace”, and do it globally

Replace
In selection

- Select your text with visual mode
- :s/find/replace/g (column s / find / replace / g)
- Find “text”, substitute it by “replace”, and do it globally

Configuration

vimrc

The config file

- Lives in ~/.vimrc
- Comments begin with “
- Documentation available online

One more thing

So much more…

- Windows
- Run commands
- Use plugins
- Write your own plugins
- And more!

Vim without vim

- Visual Studio Code
- https://marketplace.visualstudio.com/items?itemName=vscodevim.vim

- JetBrains
- https://plugins.jetbrains.com/plugin/164-ideavim

Thank you

- https://philippeloctaux.com