

Alacritty の設定

さといもちゃん

2026 年 4 月 12 日

1 はじめに

Alacritty は Rust 製ターミナルエミュレータです。Linux、macOS だけでなく、Windows でも利用可能です。ミニマム思想で設計され、例えば、画面分割やリガチャ機能などを敢えて実装していません。しかし、実装機能を最小限にとどめた恩恵で、消費リソースが少なく高速に動作します。

詳細情報は公式サイト (<https://alacritty.org>) 参照。

以降は Windows 11 環境を前提とした内容です。

2 インストール方法

Alacritty のインストールには複数の方法があります。前述の公式サイトからインストーラをダウンロードしてインストールすることも可能です。ただし、Windows 環境では WinGet を利用する方法が便利です。Windows 10 以降であれば WinGet が標準搭載されています。

WinGet を用いて Alacritty をインストールするコマンドは以下の通りです。コマンドプロンプトや PowerShell で実行可能です。

```
winget install --id Alacritty.Alacritty
```

3 設定方法

Alacritty の設定は `alacritty.toml` という単一のテキストファイルで管理します。既定の設定を `Alacritty.toml` ファイルの内容で上書きします。拡張子からわかる通り、形式は TOML です。余談ですが、TOML の LSP やフォーマッタを利用すると編集が楽になります。

初回のみ `alacritty.toml` を新規作成が必要になります。`alacritty.toml` を保存するディレクトリは以下のパスです。`xxx` はサインインしているアカウントのホームディレクトリです。

C:\Users\xxx\AppData\Roaming\alacritty

設定のリファレンスも公式サイト (<https://alacritty.org/config-alacritty.html>) に掲載されています。これを熟読することで、必要な設定を変更可能です。

4 私の設定

最後に私の設定を掲載します。

私は、カスタマイズは最小限にとどめてデフォルト設定に準じる主義です。ですので、どうしても必要な項目のみ設定しています。

```
[general]
working_directory = "c:/Users/xxx"

[window]
startup_mode = "Maximized"

[font]
size = 12
normal = { family = "JetBrainsMono NF", style = "Regular" }
bold = { family = "JetBrainsMono NF", style = "Bold" }
italic = { family = "JetBrainsMono NF", style = "Italic" }
bold_italic = { family = "JetBrainsMono NF", style = "Bold Italic" }

[terminal]
shell = "pwsh.exe"

[keyboard]
bindings = [{ key = "/", mods = "Control", chars = "\u001f" }]
```

[general] では、起動時のディレクトリのみ変更しています。毎回ディレクトリを変更する手間が煩わしいからです。

[window] では、ウィンドウが最大化された状態で起動されるように変更しています。ウィンドウが起動する時の位置やサイズが毎回異なることを避けるためです。

[font] では、フォントサイズを見やすい大きさにしています。この値はモニターにより見やすいサイズに良く変更します。また、フォントの種類を JetBrainsMono NF に変更しています。LazyVim などを利用するため Nerd Fonts 系から選んでいます。併用している WezTerm で JetBrains Mono に慣れてしまったので、Alacritty でもこのフォントに変更しています。

[terminal] では、シェルを PowerShell に変更しています。Windows 環境では pwsh.exe が私にとって最も使いやすいためです。

[keyboard] では、唯一 CTRL+/のキーバインドを設定しています。これは LazyVim 内でターミナルを起動するショートカットを機能させるための設定です。

5 おわりに

私は WezTerm という Rust 製ターミナルエミュレータと Alacritty とを併用しています。WezTerm は、タブや画面分割、リガチャ機能など多様な機能が実装されています。また、設定ファイルに Lua 言語を採用しています。このため、条件分岐を伴う設定などが容易に実現でき、カスタマイズ性が高いです。Alacritty 同様、単一テキストファイルのみで設定を一元管理可能な点も魅力的です。唯一、2024 年 2 月 3 日以降更新されていない点が気になっています。

私は汎用作業では WezTerm を利用することが多いです。ただし、高速応答が必要な場面では、Alacritty を利用しています。

Windows に標準搭載されている Windows ターミナルも意外と多機能です。少なくとも私にとっては機能の観点では必要十分です。しかし、設定ファイルを JSON ファイルで記述可能な一方で、GUI で容易に設定変更できたり、設定管理が煩わしくなったためです。現在ではほとんど利用しなくなりました。

最近では Ghostty というターミナルエミュレータにも注目しております。しかし、残念ながら、Windows 版は現時点でリリースされていません。リリースを楽しみにしています。