**Alacritty &Tmux**

Alacritty 是一个使用 OpenGL 的跨平台、GPU 加速的终端仿真器。它专注于性能和简单性，甚至没有标签或窗口拆分等功能。

Tmux 是一个用于在终端窗口中运行多个终端会话的工具，即终端复用软件（terminal multiplexer）。在 tmux 中可以根据不同的工作任务创建不同的会话，每个会话又可以创建多个窗口来完成不同的工作，每个窗口又可以分割成很多窗格。

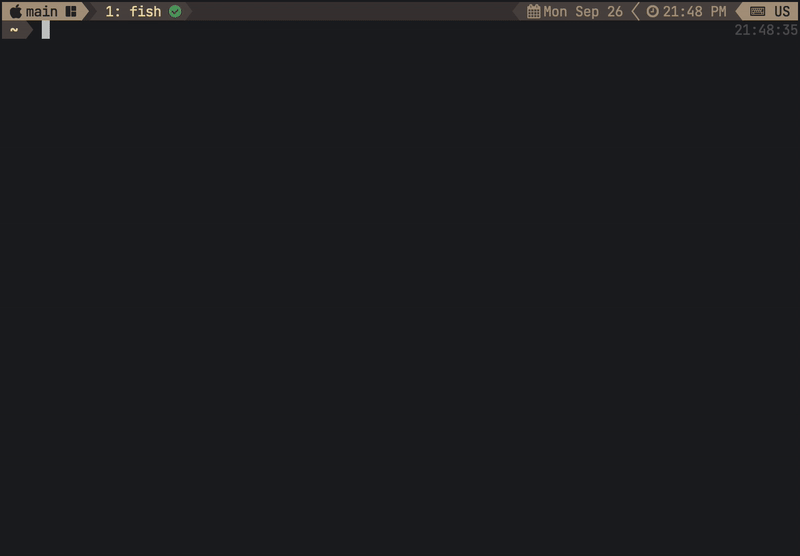
将 alacritty 和 tmux 深度集成到自己的工作流之中，打造出一个顺心应手、高效率、可定制的超级终端。

**特性**

此次深度集成，拥有 alacritty 和 tmux 各自的优点。在 macOS 完全可以代替 iTerm2.app 和 Terminal.app，达到了跨平台、高可用性、统一的超级终端。拥有以下特性：

* 快速便捷的、人体工学设计的快捷键。
* 简洁实用的、美丽大方的终端主题。
* 所有新建的窗格、选项卡使用当前的工作目录。
* 本地会话与 SSH 远程会话一视同仁。
* 自动重连的 SSH 远程会话。
* 灵活的脚本定制化。

看动图直观感受一下：



demo alacritty shortcuts

**安装**

可以通过使用 Linux、BSD、macOS 和 Windows 上的各种包管理器来安装 alacritty 和 tmux，并安装配套所需要的字体和工具。目前已在 macOS 顺畅使用，以下安装步骤以 macOS 为平台。

**Alacritty 安装**

|  |
| --- |
| Bash brew install --cask alacritty |

**Tmux 安装**

|  |
| --- |
| Bash brew install tmux |

**字体安装**

我选择打过 Nerd Font 字体补丁的 JetBrains Mono 作为终端的默认字体。

|  |
| --- |
| Bash brew tap homebrew/cask-fonts brew install font-jetbrains-mono brew install font-jetbrains-mono-nerd-font |

**Fish 安装**

我选择 fish 作为终端的默认 Shell。（可选，可修改配置使用其他 Shell）

|  |
| --- |
| Bash brew install fish |

**Fzf 安装**

快速模糊查找。（可选，仅在命令菜单和会话菜单中使用）

|  |
| --- |
| Bash brew install fzf |

**Autossh 安装**

自动重连的 SSH 客户端。（可选，可以修改配置使用普通的 SSH 客户端）

|  |
| --- |
| Bash brew install autossh |

**配置**

**Alacritty 配置**

Alacritty 默认使用的配置文件路径 ~/.config/alacritty/alacritty.yml。 我正在使用的配置文件：[alacritty.yml](https://link.zhihu.com/?target=https%3A//github.com/qingshan/dotfiles/blob/main/alacritty/alacritty.yml)。 运行下面的命令可以把我的配置文件自动下载并使用：

|  |
| --- |
| Bash curl -fLo ~/.config/alacritty/alacritty.yml --create-dir \  https://raw.githubusercontent.com/qingshan/dotfiles/main/alacritty/alacritty.yml |

**Tmux 配置**

Tmux 默使用的配置文件路径 ~/.tmux.conf 我正在使用的配置文件：[.tmux.conf](https://link.zhihu.com/?target=https%3A//github.com/qingshan/dotfiles/blob/main/.tmux.conf)。 运行下面的命令可以把我的配置文件自动下载并使用：

|  |
| --- |
| Bash curl -fLo ~/.tmux.conf \  https://raw.githubusercontent.com/qingshan/dotfiles/main/.tmux.conf |

**快捷键冲突**

绑定的快捷键与 alacritty 和 macOS Screenshot 默认的一些快捷键冲突，我选择调整冲突的这些默认快捷键。

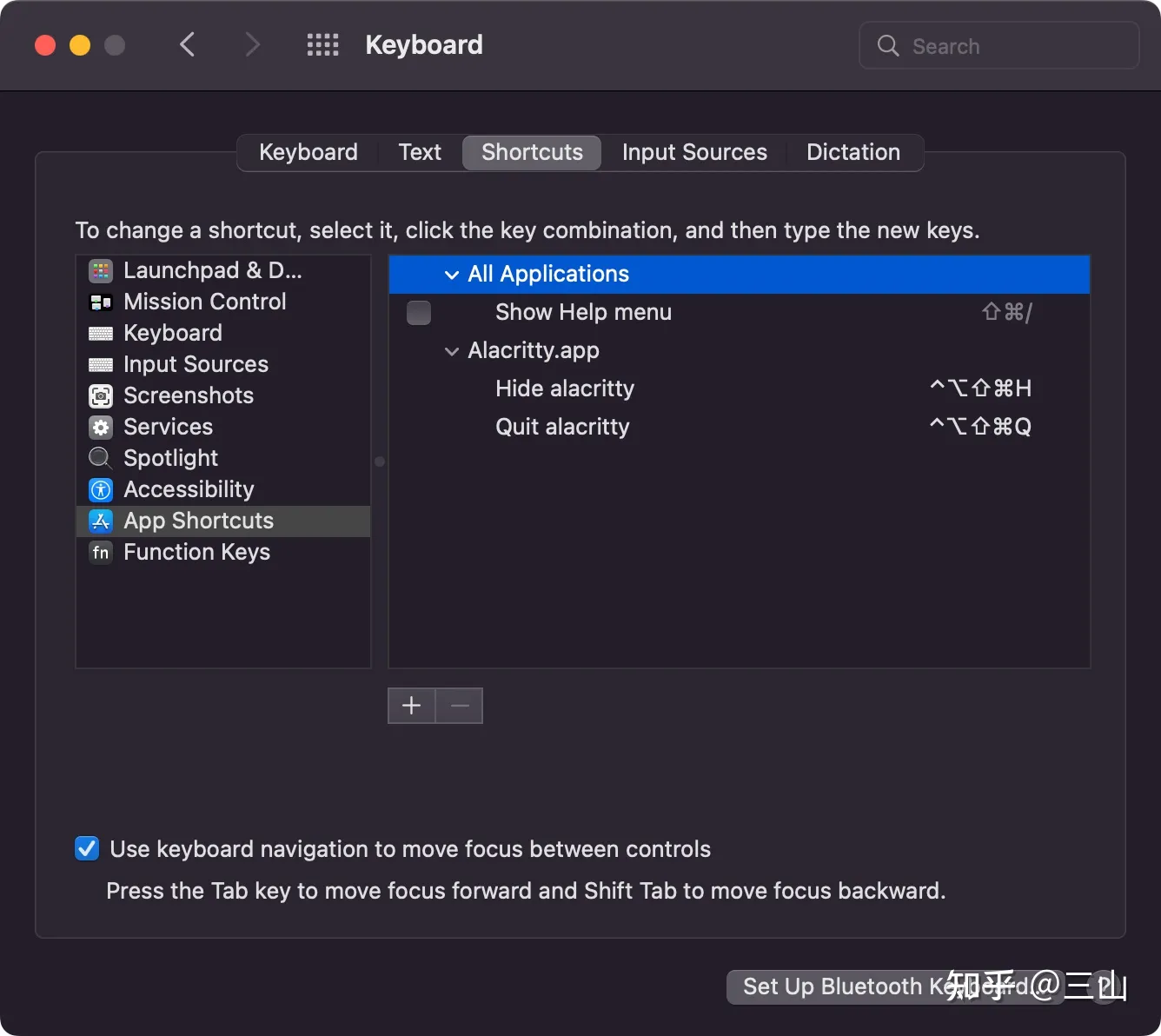
* Alacritty 有两个快捷键冲突：Command + H 和 Command + Q；
* macOS Screenshot 有多个数字快捷键冲突：Command + Shift + 3 到 6

**Alacritty 快捷键冲突**

在 System Preferences / Keyboard / Shortcuts / App Shortcuts 下点击 + 按钮，添加两项菜单主题对应的快捷键：

* Hide alacritty
* Quit alacritty

定义的快捷键只要不是冲突的那两个快捷键都可以，添加后的结果如下：



remap\_alacritty\_shortcuts

**macOS Screenshot 快捷键冲突**

在 System Preferences / Keyboard / Shortcuts / Screenshots 下取消所有 Screenshot 的快捷键。因为我可以直接运行 Screenshot.app 来截图。

**使用**

绑定快捷键使用Command或者Command + Shift作为修饰键。部分快捷键与 iTerm2.app 和 Terminal.app 的绑定对应，部分快捷键与我的 i3 的绑定对应。

**基本操作**

新建

* Command + D 垂直分割窗格
* Command + Return 水平分割窗格
* Command + T 新建选项卡

关闭

* Command + W 关闭窗格
* Command + Shift + W 关闭选项卡
* Command + Q 关闭窗口

访问窗格

* Command + H 或者 Command + Left 访问左边的窗格
* Command + J 或者 Command + Up 访问上面的窗格
* Command + K 或者 Command + Down 访问下面的窗格
* Command + L 或者 Command + Right 访问右边的窗格

访问选项卡

* Command + 1 到 Command + 9 - 按数字切换选项卡
* Command + b 切换到最近一次访问的选项卡
* Command + [ 切换到上一个选项卡
* Command + ] 切换到下一个选项卡

访问窗口

* Command + `：切换窗口。

**布局操作**

调整窗格大小

* Command + Shift + H 或者 Command + Shift + Left 向左边的窗格推进
* Command + Shift + J 或者 Command + Shift + Up 向上面的窗格推进
* Command + Shift + K 或者 Command + Shift + Down 向下面的窗推进
* Command + Shift + L 或者 Command + Shift + Right 向右边的窗格推进

移动窗格到选项卡

* Command + Shift + 1 到 9: 将窗格移动到指定的选项卡中。

缩放窗格

* Command + Z 缩放当前窗格。

调整窗格布局

* Command + Shift + Z：使用预设置的五种布局重新调整窗格。

修改选项卡名称

* Command + ,：修改选项卡名称。

**广播输入**

* Command + Shift + I：广播输入到当前选项卡的所有窗格。

**Tmux 命令**

* Command + I：输入 tmux 命令。

**字体操作**

* Command + +：调整更大的字体
* Command + -：调整更小的字体
* Command + 0：恢复默认大小的字体

**复制模式**

**Vi 复制模式**

查找关键词并进入 Vi 复制模式：

* Command + F：进入 Vi 复制模式，从上往下的方向查找关键词。
* Command + Shift + F：进入 Vi 复制模式，从下往上的方向查找关键词。

进入 Vi 复制模式后：

* v 按字选择
* V 按行选择
* Ctrl + v 按块状选择
* Esc 取消选择
* A 复制选择的文本追加到剪贴板，并退出 Vi 复制模式。
* D 复制当前行到行尾的文本到剪贴板，并退出 Vi 复制模式。
* y 复制选择的文本到剪贴板，并退出 Vi 复制模式。
* Y 复制选择的文本到剪贴板，并保持 Vi 复制模式。
* q 退出 Vi 复制模式。
* 还有更多，请查文档。

**鼠标模式**

使用鼠标选择文本，会自动复制到操作系统的剪贴板。

**进阶**

**快捷命令**

* Command + R: 弹出快捷命令菜单，并在当前窗格执行选择的命令。
* Command + Shift + D：弹出快捷命令菜单，并在新建垂直分割的窗格执行选择的命令。
* Command + Shift + Retrun：弹出快捷命令菜单，并在新建水平分割的窗格执行选择的命令。
* Command + Shift + T：弹出快捷命令菜单，并在新建的选项卡执行选择的命令。

快捷命令弹出一个窗口，可以选择需要执行的命令。快捷命令列表由自定义一个可执行的脚本文件 tmux-commands 负责提供。脚本文件放到 PATH 路径中。 可以参考我的脚本文件：[tmux-commands](https://link.zhihu.com/?target=https%3A//raw.githubusercontent.com/qingshan/dotfiles/main/bin/tmux-commands)。 脚本文件内容如下（命令列表内容根据需要自行修改）：

|  |
| --- |
| Bash #!/usr/bin/env bash (fzf --reverse --header Commands) <<EOF cargo run cargo test ssh lab EOF |

直接执行这个命令确认一下：

|  |
| --- |
| Bash tmux-commands |

从快捷命令中选择一个命令，会直接输出打印出来。确认没有问题之后，直接通过上面的快捷键呼叫。

**快捷会话**

* Command + N：弹出窗口显示本地会话列表，并在当前窗口切换到选择的会话中。
* Command + Shift + N：弹出窗口显示本地会话和远程会话列表，并新建一个窗口切换到选择的会话中。

快捷会话默认从 tmux 加载会话列表。也可以自定义一个可执行的脚本文件：tmux-sessions，放到 PATH 路径中。 可以参考我的脚本文件：[tmux-sessions](https://link.zhihu.com/?target=https%3A//raw.githubusercontent.com/qingshan/dotfiles/main/bin/tmux-sessions)。

注意：远程服务器上也应该有 tmux-commands 和 tmux-sessions 两个脚本文件。

**保存文件**

* Command + S：输出 Vim 保存文件的指令 <ESC>:w<CR>，注意要在 Vim 下使用。

时不时的保存文件是个好习惯。

**Tmux 快捷键**

因为通常 tmux 绑定的快捷键前缀 Ctrl-a 和 Ctrl-b 经常在 Shell 或者 Vim 中有使用，所以我重新定义的 tmux 的 快捷键前缀为 Ctrl + \。 选择的原因是这个快捷键很少被其他应用程序使用，另外一个原因是在我的集成中没有太多机会使用 tmux 的快捷键，所以选择一个不冲突的快捷键前缀对我来说是最好的选择。

**配置说明**

**配置终端类型**

给 alacritty 追加环境变量：

|  |
| --- |
| YAML env:  TERM: xterm-256color |

**配置字体**

指定使用 JetBrains Mono 字体，字体大小设置为 14 点数。

|  |
| --- |
| YAML font:  normal:  family: JetBrainsMono Nerd Font  style: Regular   bold:  family: JetBrainsMono Nerd Font  style: Bold   italic:  family: JetBrainsMono Nerd Font  style: Italic   bold\_italic:  family: JetBrainsMono Nerd Font  style: Bold Italic   size: 14 |

**配置窗口**

Alacritty 启动的时候使用全屏模式，但是又不独占一个桌面。窗口不需要任何裱装。

|  |
| --- |
| YAML window:  startup\_mode: SimpleFullscreen  decorations: none |

**配置 Shell**

使用 Fish 作为默认的 Shell。启动的时候总是直接创建或使用名为 main 的 tmux 会话。

|  |
| --- |
| Shell shell:  program: /usr/local/bin/fish  args:  login  command  "tmux new-session -A -D -s main" |

**配置快捷键映射**

可以映射到 tmux 的绑定：

|  |
| --- |
| Shell { key: T, mods: Command, chars: "\x1cc" } *new tab with default shell* |

甚至 Vim 指令：

|  |
| --- |
| Shell { key: S, mods: Command, chars: "\x1b:w\x0a"} *Type <escape>:w<enter> to save vim* |

还可以可以映射运行程序：

|  |
| --- |
| Shell { key: N, mods: Command|Shift, command: { program: "/usr/local/bin/alacritty", args: ["msg", "create-window", "-e", "/usr/local/bin/fish", "--login", "--command", "tmux-sessions --all"] } } *open session in new window with all sessions menu across servers* |

**奖励**

**Alacritty 图标**

Alacritty 默认的图标在 Dock 上与其他应用程序的图标有点格格不入。但是作者不愿意调整图标上，那我只好自己动手调整。 参考：[issue:3926](https://link.zhihu.com/?target=https%3A//github.com/alacritty/alacritty/issues/3926) 下载 [Alacritty.icns](https://link.zhihu.com/?target=https%3A//www.dropbox.com/s/0i4ez0el7paksg3/Alacritty.icns%3Fdl%3D0) 并更新 Alacritty.app

**输入法退格键**

macOS 下输入法退格键会删除已输入内容的问题：[issues:1606](https://link.zhihu.com/?target=https%3A//github.com/alacritty/alacritty/issues/1606)

官方给出的临时解决方案为：

|  |
| --- |
| Shell key\_bindings:  { key: Back, action: ReceiveChar } |

**Alt 组合键映射**

所有以 Alt 和 Alt + Shift 为修饰键的按键都映射为 Esc 前缀键码。

**IDE 布局**

写一个脚本并执行实现自动创建类似 IDE 布局：

|  |
| --- |
| Bash tmux split-window -v -p 30 tmux send-keys "echo 'run'" enter tmux split-window -h -p 50 tmux send-keys "echo 'preview'" enter |

**后续**

* [ ] 加上 tmux-resurrect 插件，断电或者电脑重启后，都可以快速进入基本工作环境。
* [ ] 使用 tmux display-popup 实现快捷方便的弹出窗口。快捷键可以考虑映射 Command + F1 到 F12。

**参考资料**

* [Alacritty integration with Tmux](https://link.zhihu.com/?target=https%3A//arslan.io/2018/02/05/gpu-accelerated-terminal-alacritty/)
* [macOS Keyboard Shortcuts for tmux](https://link.zhihu.com/?target=https%3A//www.joshmedeski.com/posts/macos-keyboard-shortcuts-for-tmux)
* [Alacritty's documentation](https://link.zhihu.com/?target=https%3A//github.com/alacritty/alacritty/blob/master/docs/features.md)
* [tmux Manual Pages](https://link.zhihu.com/?target=https%3A//man7.org/linux/man-pages/man1/tmux.1.html)