

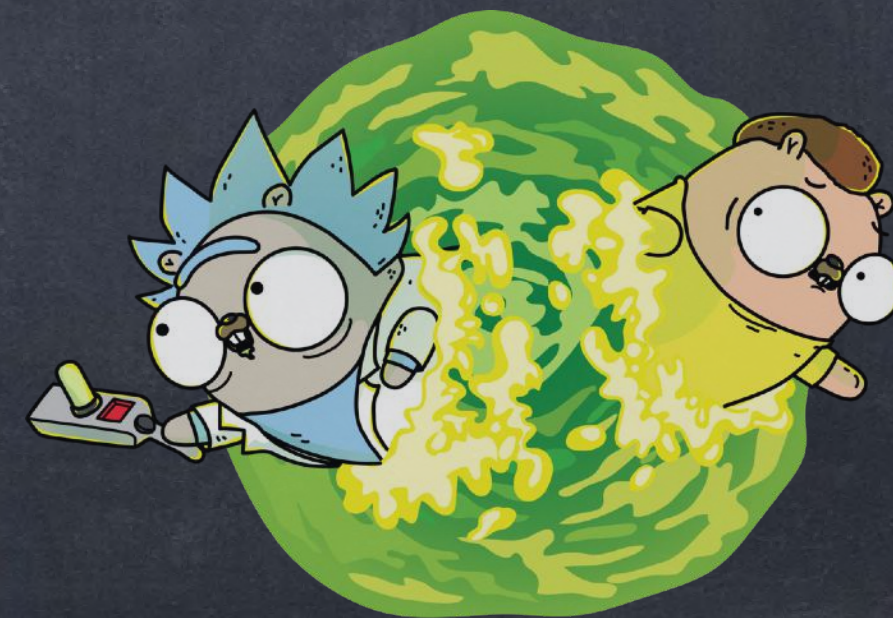
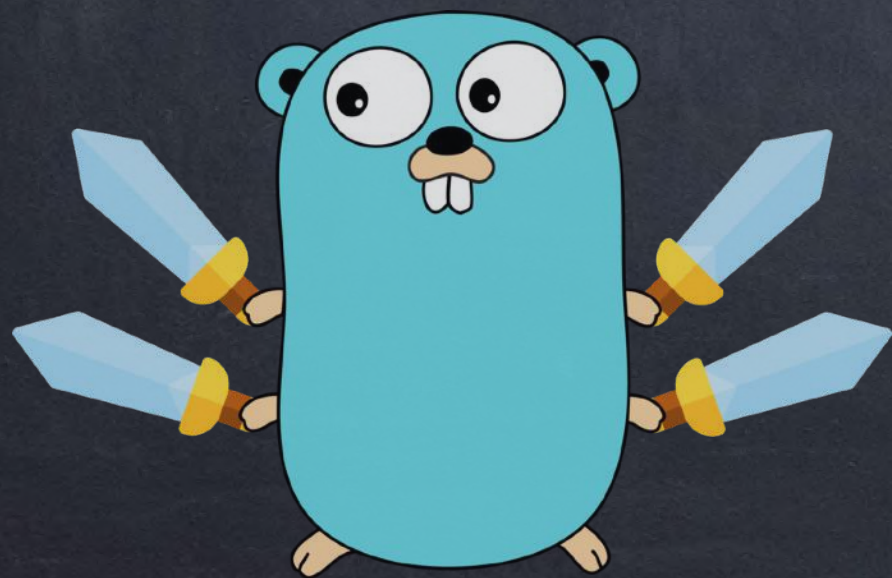
v3.1

优雅的编程者

between a and c

– xiaorui.cc

– github.com/rfyiamcool



先从xx windows开始...

- 👁 mac
- 👁 ubuntu
- 👁 manjaro (arch linux)



menu

- 我们的目标

- 高效编程

- 好看绚丽

- 保护肾腰

- 应用篇

- 终端篇

- 设备篇

- 终极篇

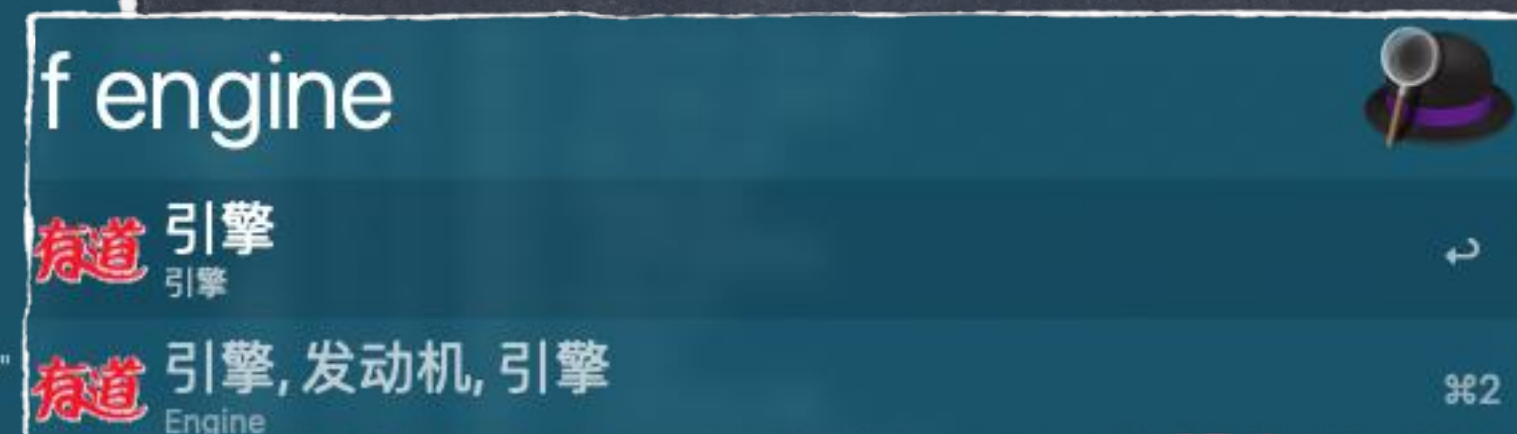
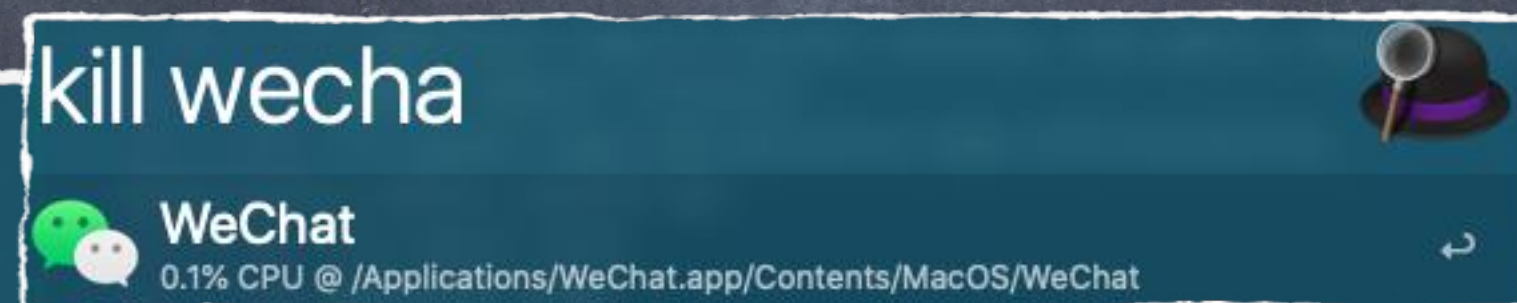
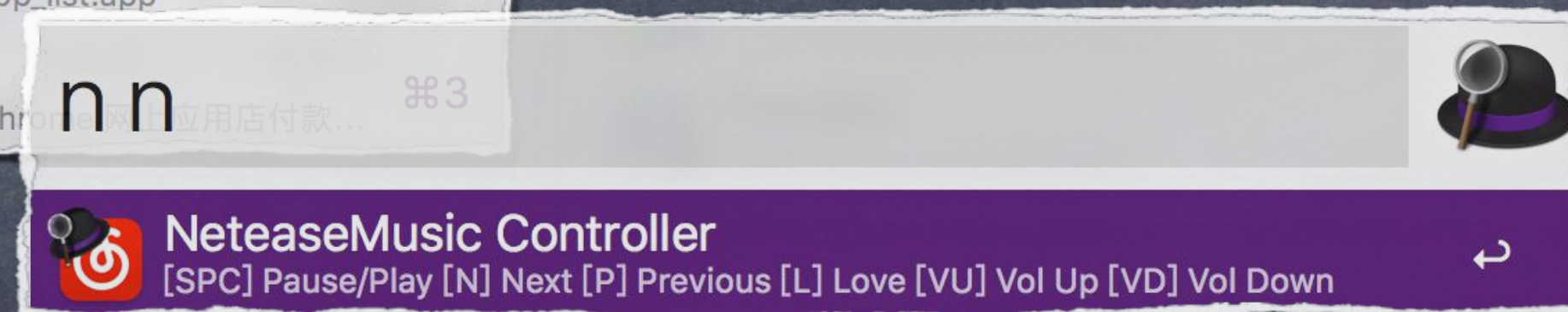
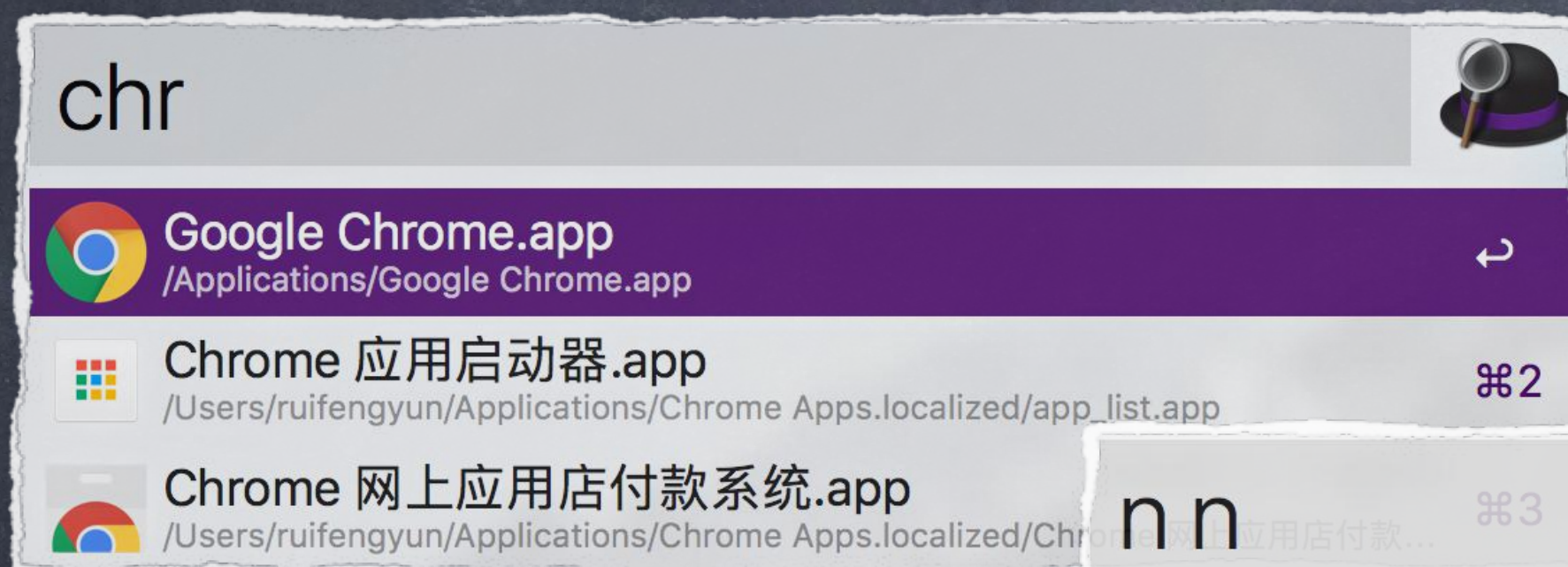


应用篇

- chrome
- alfred 快捷管理
- notion && evernote
- hyperSwith & contexts
- iterm2 终端
- vscode 通用编辑器
- neovim 定制编辑器
- sketch 画图
- mindnode 思维导图
- dash 文档
- magnet 分屏
- karabiner 键位映射
- clashx 你懂的



alfred



自定义workflow

拉起进程

切换进程

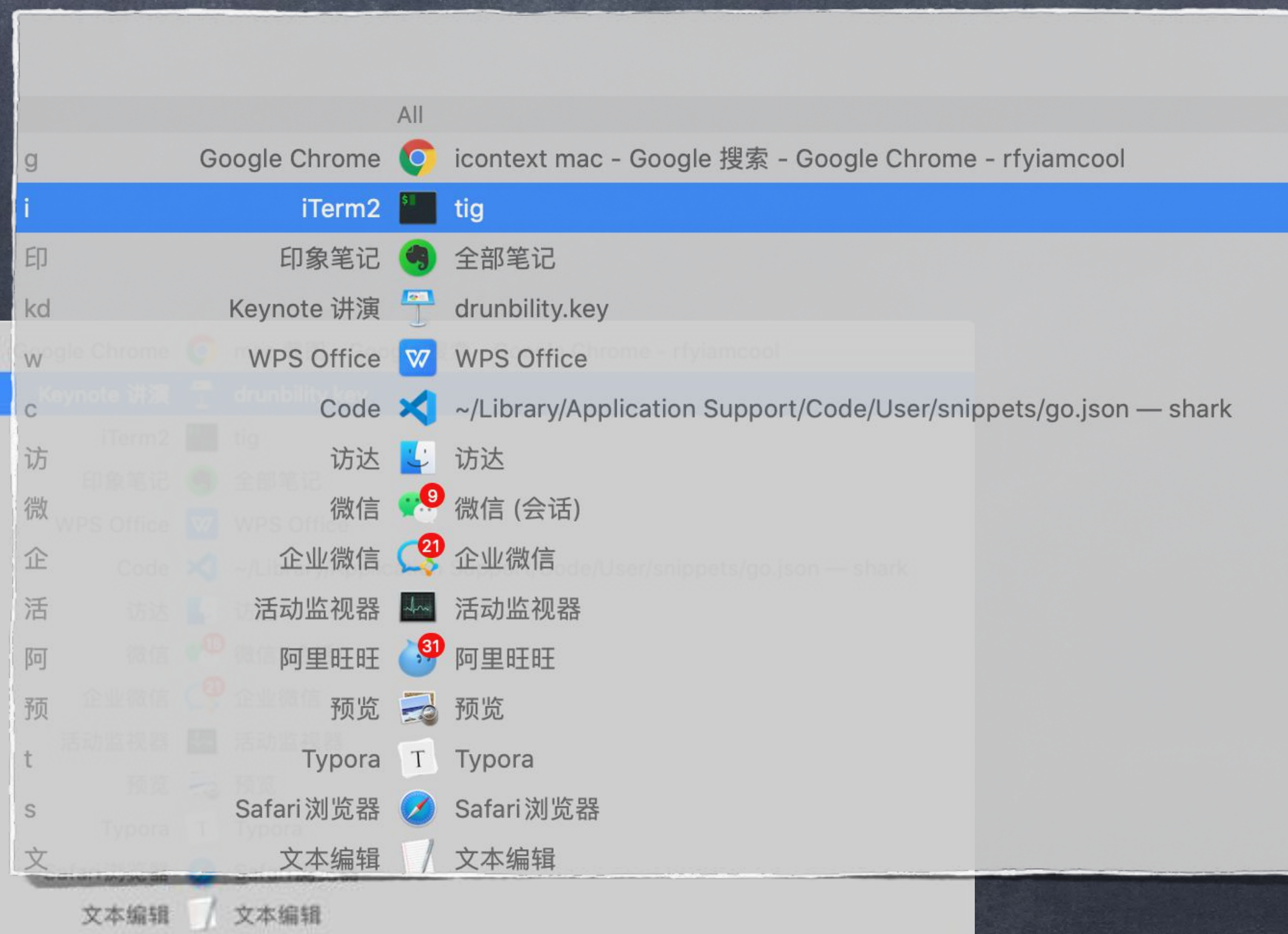
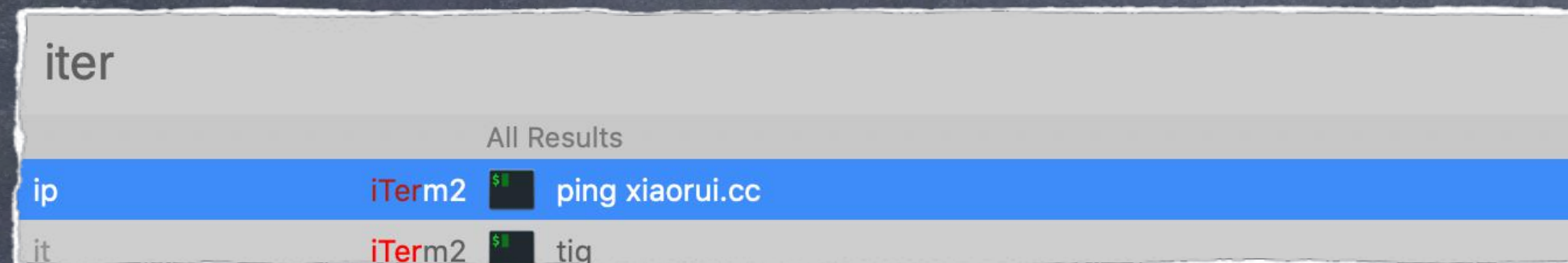
各种搜索

定制脚本

粘贴历史

...

contexts



窗口管理器

热度排序

区分相同应用

搜索匹配

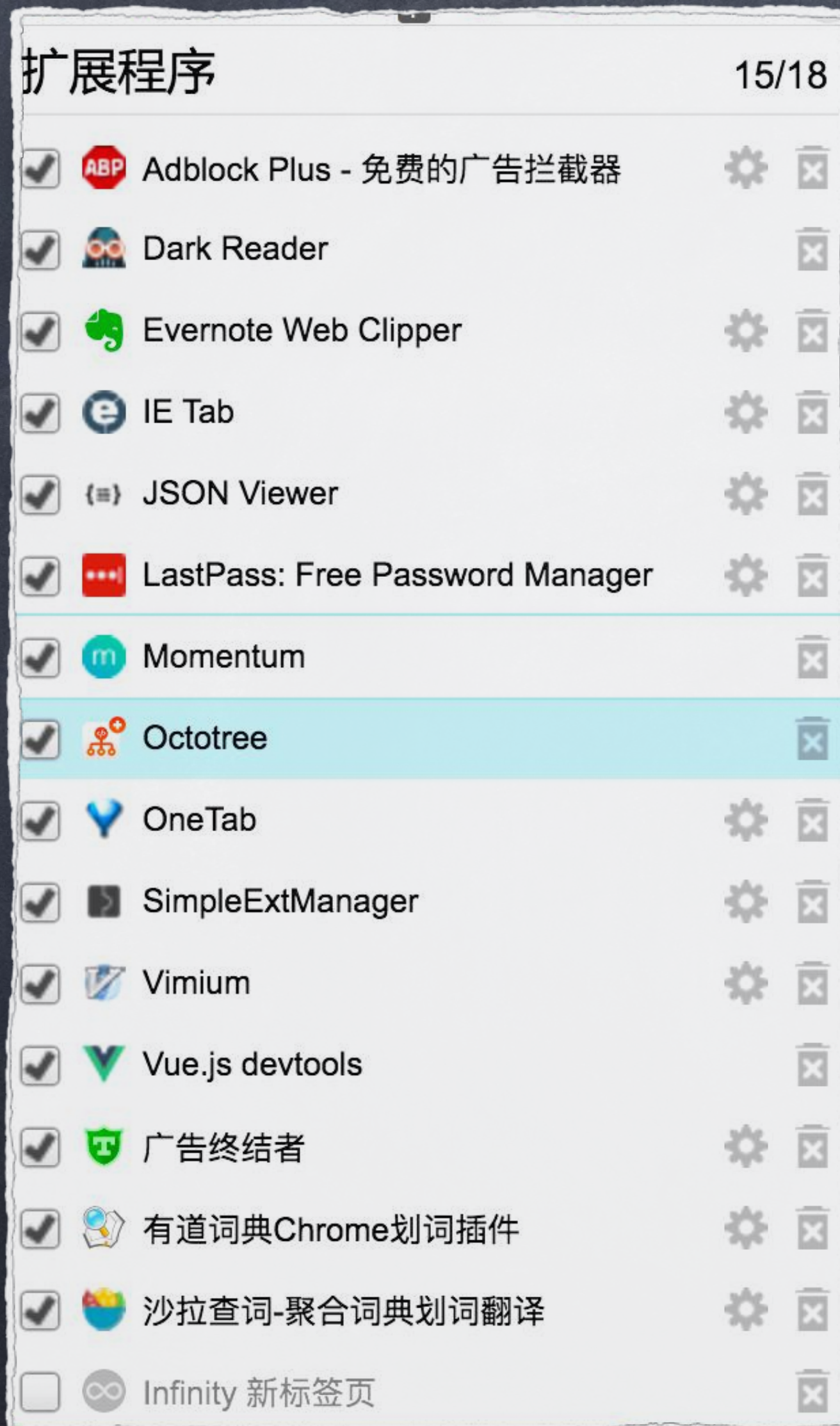
索引号

字符串

劫持默认tab

...

chrome

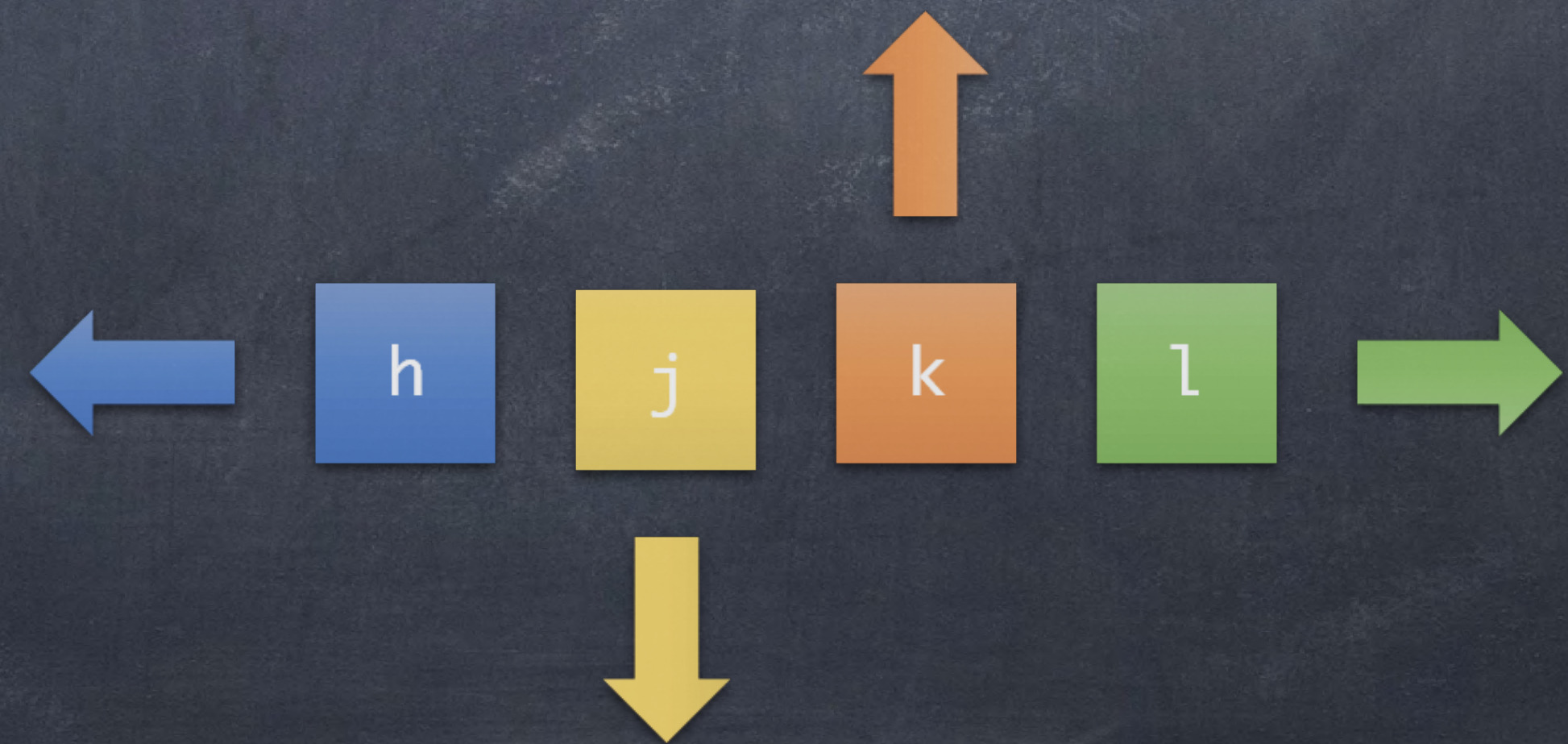


vimium & Octotree

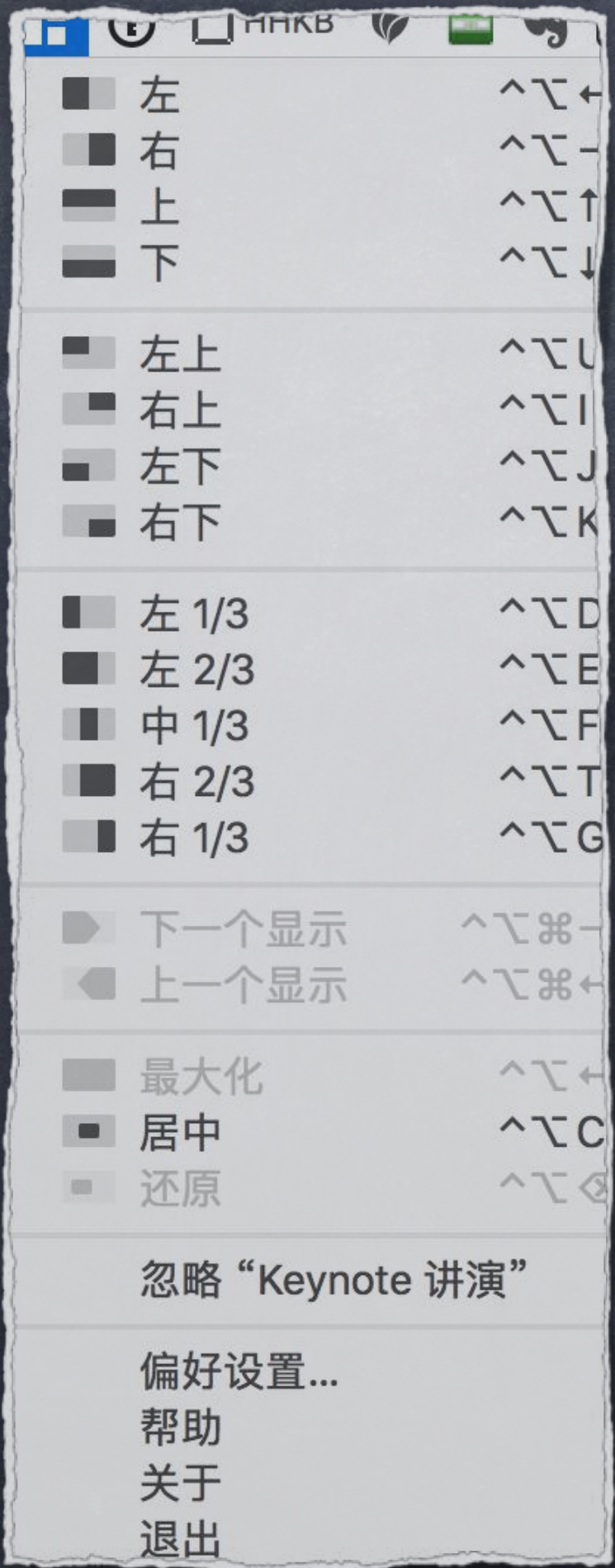
karabiner

- 把caps_lock改成control键位
- caps_lock属于“非”常用键
- 左下角control引起小拇指
- 改其他组合键为caps_lock

- 为什么要改方向键 !!!
- 方向键通常在右下角, 所以每次都要抬手移过去...
- 上下左右 = control + kjhl
- 同理改 pageup/pagedown



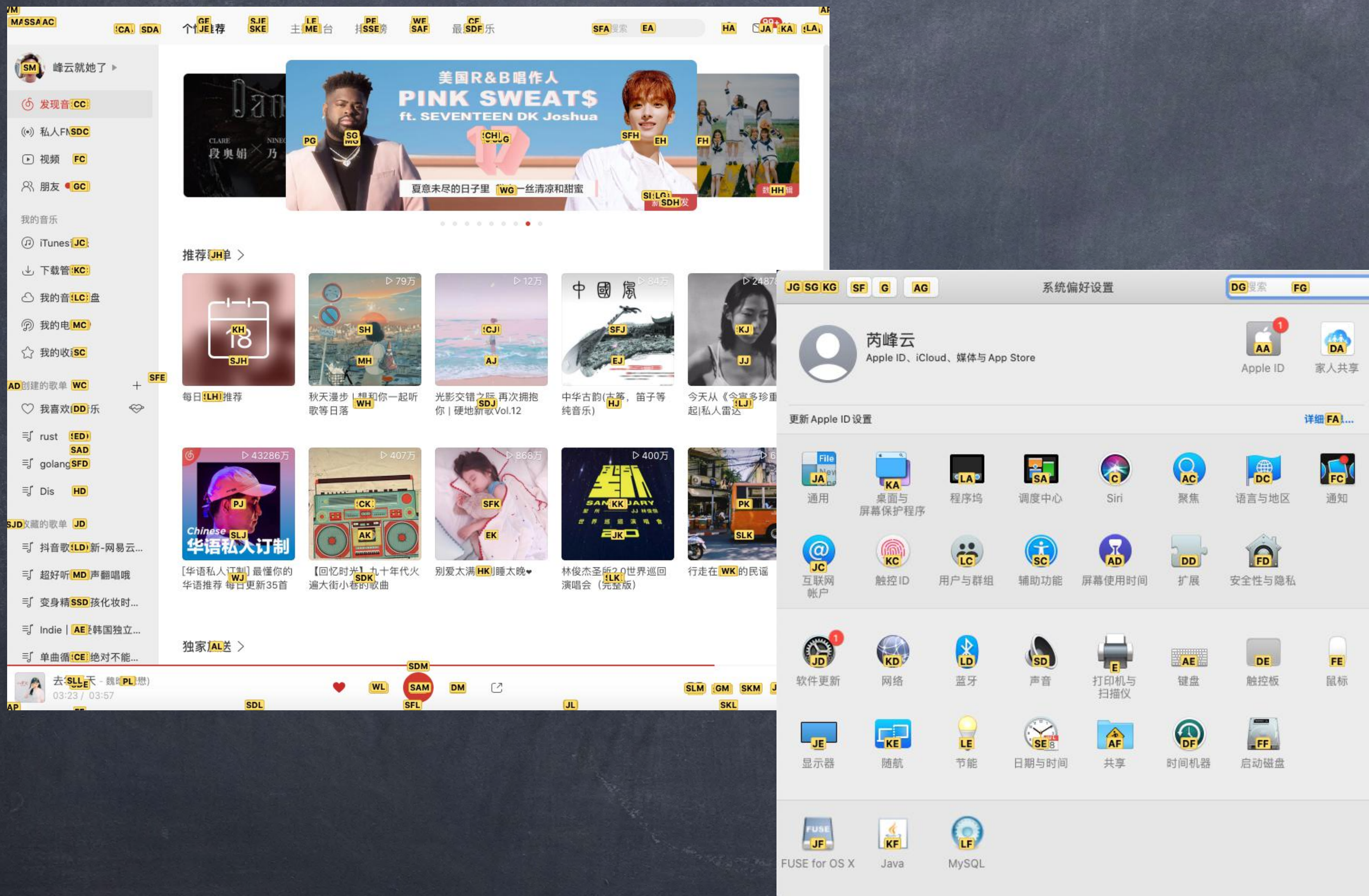
高效窗口管理



- 窗口尺寸变更
- 多屏幕之间转移
- 窗口对其
- ...

这类工具很多...

vimac



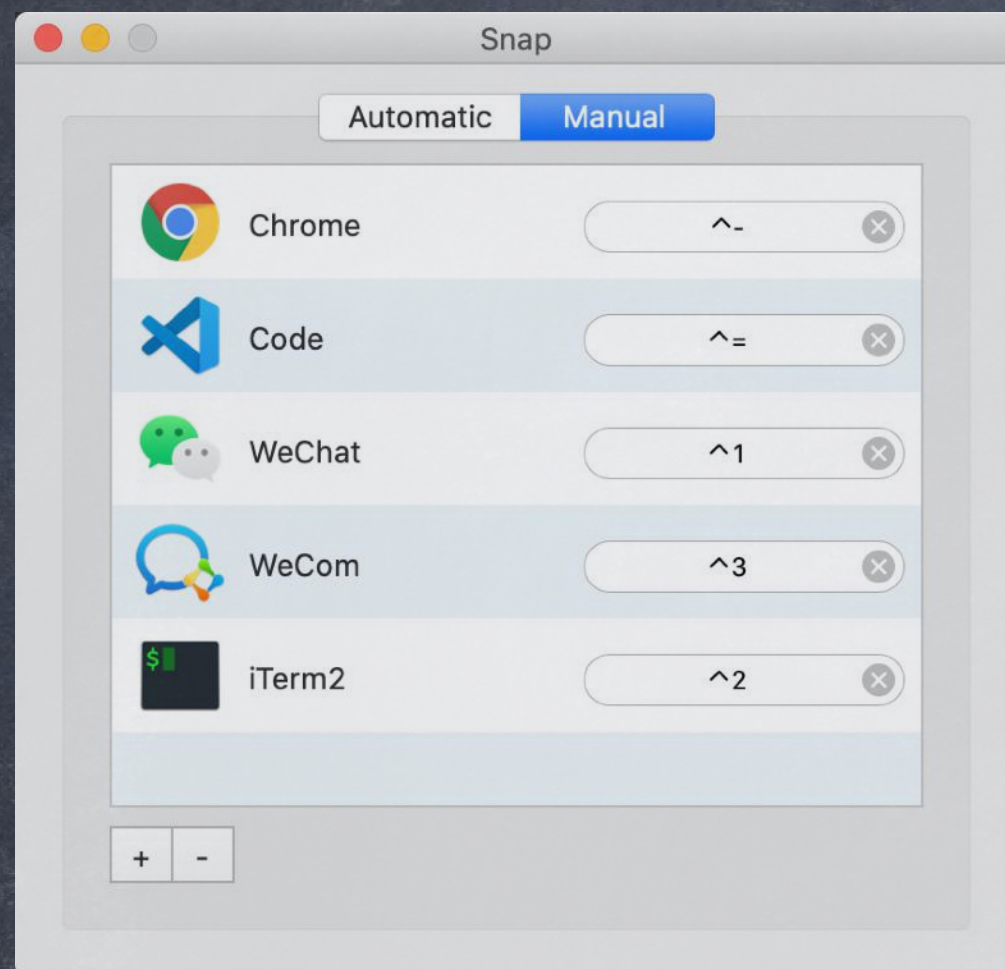
mac + vim

展示快捷键

滚动屏幕

...

other

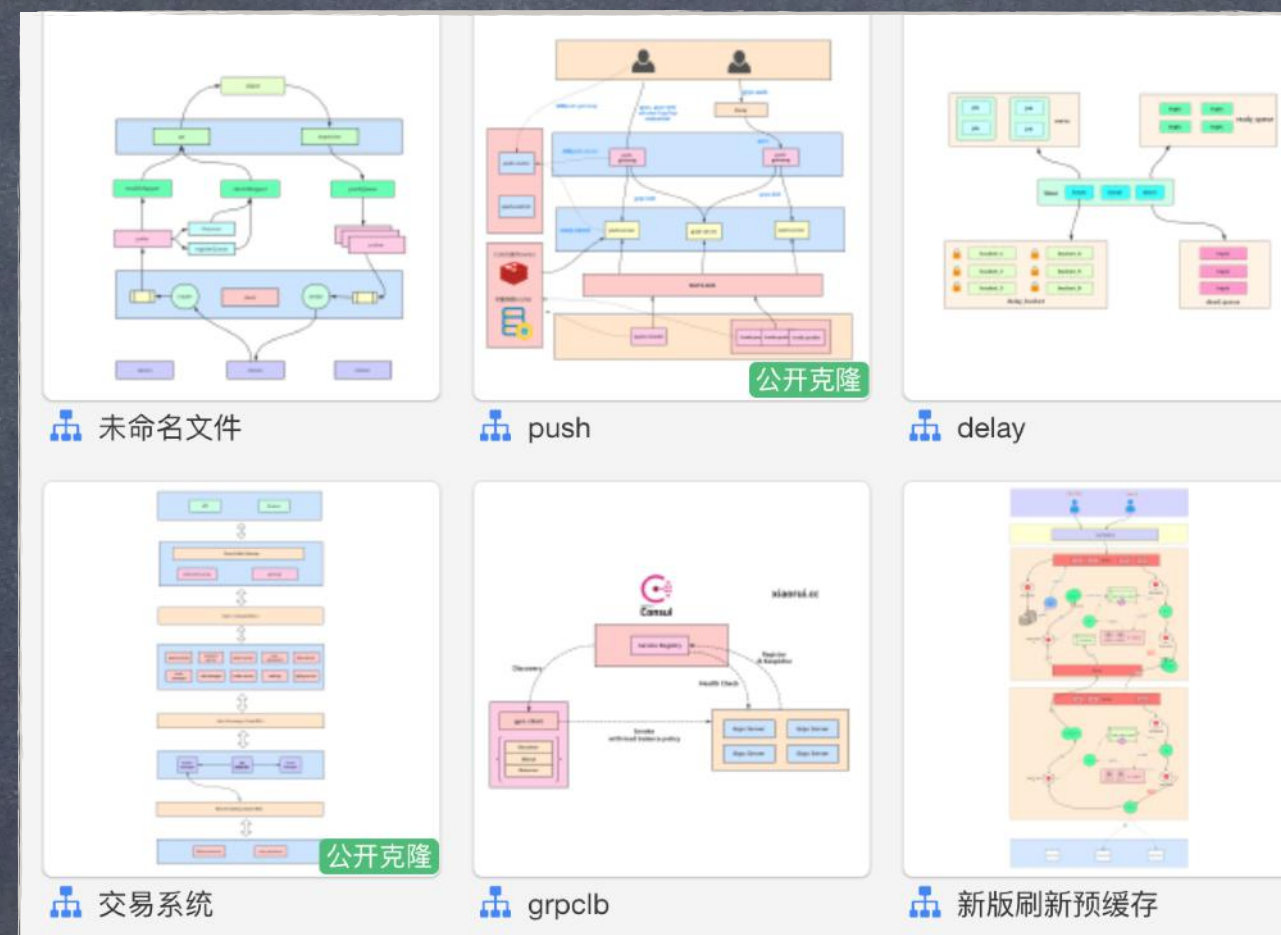


👁️ snap

👁️ 一键到达

👁️ 一键切换

👁️ / ... /



👁️ 架构图

👁️ processon.com

👁️ [omniGraffle](https://omniGraffle.com)

👁️ ppt

👁️ yanshuo.io

👁️ slides.com

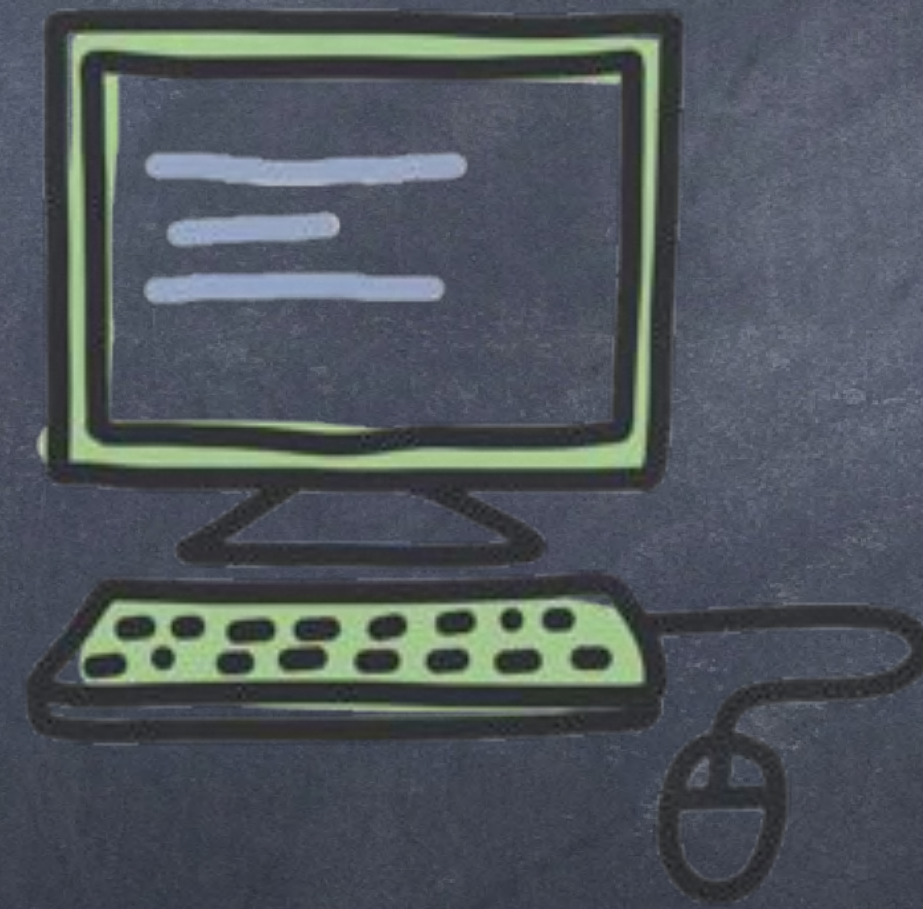
👁️ reveal.js

你好，世界

@xiaorui.cc

终端篇

- `iterm2`
- `zsh`
- `tmux` & `tmuxp`
- `vim`
- 好用的命令
- 快捷键



iterm2

- iterm2

- font

- nerd-fonts

- theme

- dracula (current theme)

- molokai

- solarized dark || light

```
~/go/src/github.com/rfyiamcool/grpcall  master ?  
$ ll  
total 224  
rwxr-xr-x  16 ruifengyun  staff    512   3   2 00:31 .git  
rw-r--r--   1 ruifengyun  staff   2599   8  14 2019 README.md  
rw-r--r--   1 ruifengyun  staff  12678   3  18 2019 enter.go  
rw-r--r--   1 ruifengyun  staff   1610   3  18 2019 types_cache.go  
rw-r--r--   1 ruifengyun  staff    932   3  18 2019 common.go  
rw-r--r--   1 ruifengyun  staff  19235   3  17 2019 core.go  
rwxr-xr-x   9 ruifengyun  staff    288   3  17 2019 testing  
rw-r--r--   1 ruifengyun  staff   8569   3  17 2019 desc_source.go  
rw-r--r--   1 ruifengyun  staff  11744   3  15 2019 invoke.go  
rw-r--r--   1 ruifengyun  staff   3395   2  25 2019 format.go  
rw-r--r--@  1 ruifengyun  staff  20388   2  21 2019 design.jpg  
rw-r--r--   1 ruifengyun  staff   4721   2  21 2019 Gopkg.lock  
rw-r--r--   1 ruifengyun  staff    956   2  21 2019 Gopkg.toml  
rw-r--r--   1 ruifengyun  staff     31   2  19 2019 .gitignore  
  
~/go/src/github.com/rfyiamcool/grpcall  master ?  
$
```

iterm2

- 👁 最前 & 最后

- 👁 control + a

- 👁 control + b

- 👁 单词删

- 👁 option + backspace

- 👁 历史

- 👁 shift + cmd + h

- 👁 搜索终端

- 👁 shift + cmd + o



- 👁 单词移动

- 👁 option + b

- 👁 option + f

- 👁 iterm2 窗口跳转

- 👁 command + 左右

iterm2 (zsh + theme)

```
~/go/src/github.com/rfyiamcool/grpcall  🐱  master ?  
$ aa  
zsh: command not found: aa  
  
~/go/src/github.com/rfyiamcool/grpcall  🐱  master ?  
$ kkk  
zsh: command not found: kkk  
  
~/go/src/github.com/rfyiamcool/grpcall  🐱  master ?  
$ ssh vps.xiaorui.cc -l root
```

theme

- powerlevel9k
- powerlevel10k (async)
- pure (async)
- spaceship

plugin

- git & vi-mode
- autojump
- zsh-syntax-highlighting
- zsh-autosuggestions

zsh plugin vi-mode

```
$ curl 'https://dss0.baidu.com/73F1bjeh1BF3odCf/it/u=3624547722,2465723046&fm=85&s=63f8A87206B66E25451477460300E0F5' -H 'authority: dss0.baidu.com' -H 'user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/537.36' -H 'sec-fetch-dest: image' -H 'accept: image/webp,image/apng,image/*,*/*;q=0.8' -H 'sec-fetch-site: same-site' -H 'sec-fetch-mode: no-cors' -H 'referrer: https://www.baidu.com/s?wd=mm&pn=20&oq=mm&tn=baiduhome_pg&ie=utf-8&usm=4&rsv_idx=2&rsv_pq=e233615400014ba3&rsv_t=da32cXXE6ePHEqEzvydJPqapaj7Sf5kuMF%2BEqlttZ%2FJf0lGolXYy2Ken7ja7X7%2F35ewJI&rsv_page=1' -H 'accept-language: zh-CN,zh;q=0.9,en;q=0.8,zh-TW;q=0.7' -H 'cookie: PSTM=148420423; BIDUPSID=205A550BB27CF17CBA71F; BAIDUCUID=++; BAIDUID=CCF36181DD304D41E58DEE2BD377D:FG=1; BDUSS=Ek2R3RZTmpIVHZRekl4VjV3SWlMNC1qb1Q0VldlVmFVdzU5LWwya29SRzVIU2hlRVFBQUFBQAAAAAAAAAAAAAAADt5jA53Me35dTGNTIxAAAAAAAAAAAAAAAAAAALmQAF65kABed; MCITY=-%3A; BDRCVFR[feWj1Vr5u3D]=I67x6TjHwwYf0; delPer=0; PSINO=2; BDORZ=B490B5EBF6F3CD402E515D22BCDA1598; H_PS_PSSID=1422_21122_30906_30824_22157' --compressed
```

👁 **esc**

👁 用**vim**的模式操作命令行

👁 **esc + v**

👁 导入到**vim**里再去操作命令

```
1 curl 'https://dss0.baidu.com/73F1bjeh1BF3odCf/it/u=3624547722,2465723046&fm=85&s=63f8A87206B66E25451477460300E0F5' -H 'authority: dss0.baidu.com' -H 'user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/537.36' -H 'sec-fetch-dest: image' -H 'accept: image/webp,image/apng,image/*,*/*;q=0.8' -H 'sec-fetch-site: same-site' -H 'sec-fetch-mode: no-cors' -H 'referrer: https://www.baidu.com/s?wd=mm&pn=20&oq=mm&tn=baiduhome_pg&ie=utf-8&usm=4&rsv_idx=2&rsv_pq=e233615400014ba3&rsv_t=da32cXXE6ePHEqEzvydJPqapaj7Sf5kuMF%2BEqlttZ%2FJf0lGolXYy2Ken7ja7X7%2F35ewJI&rsv_page=1' -H 'accept-language: zh-CN,zh;q=0.9,en;q=0.8,zh-TW;q=0.7' -H 'cookie: PSTM=148420423; BIDUPSID=205A550BB27CF17CBA71F; BAIDUCUID=++; BAIDUID=CCF36181DD304D41E58DEE2BD377D:FG=1; BDUSS=Ek2R3RZTmpIVHZRekl4VjV3SWlMNC1qb1Q0VldlVmFVdzU5LWwya29SRzVIU2hlRVFBQUFBQAAAAAAAAAAAAAAADt5jA53Me35dTGNTIxAAAAAAAAAAAAAAAAAAALmQAF65kABed; MCITY=-%3A; BDRCVFR[feWj1Vr5u3D]=I67x6TjHwwYf0; delPer=0; PSINO=2; BDORZ=B490B5EBF6F3CD402E515D22BCDA1598; H_PS_PSSID=1422_21122_30906_30824_22157' --compressed
```

fzf

智能历史记录查询

command + r

```
~/go/src/github.com
$
9365 go14 run dd.go
9366 go run
9389 vim runtime2.go
9417 vim src/runtime/panic.go
9425 go14 test -bench=. -run=none
9427 go test -bench=. -run=none
9453 vim runtime/panic.go
9456 vim /Users/ruifengyun/bin/go/src/runtime/panic.go
9465 go test -bench=. -run=none -benchmem .
9466 go14 test -bench=. -run=none -benchmem .
9471 go run a.go
9472 go14 run a.go
> 9596 go run main.go
1124/10000 +S
> go run
```

更快可模糊的文件搜索

option + c

```
~/go/src/github.com
$
> github.com rfyiamcool grpc
300/17953
> rfyiamcool/grpc-proxy/vendor/github.com
rfyiamcool/istio_grpc/vendor/github.com
rfyiamcool/grpc-proxy_2/vendor/github.com
rfyiamcool/grpc-sidecar/vendor/github.com
rfyiamcool/grpc-proxy/vendor/github.com/rs
rfyiamcool/grpc-proxy/vendor/github.com/pkg
rfyiamcool/grpc-proxy/vendor/github.com/gogo
rfyiamcool/grpc-proxy_2/vendor/github.com/rs
rfyiamcool/grpc-sidecar/vendor/github.com/rs
rfyiamcool/grpc-proxy_2/vendor/github.com/pkg
rfyiamcool/grpc-sidecar/vendor/github.com/pkg
rfyiamcool/grpc-proxy/vendor/github.com/beorn7
```

fzf

路径使用 ** 可模糊查询

```
~/go/src/github.com/rfiamcool
$ cd **
> grpc
936/2207
grpcx
grpcpp
> grpcall
grpc-proxy
grpc-codec
istio_grpc

~/go/src/github.com
$ vim rfiamcool/**
8362/8362
rfiamcool
rfiamcool/backoff
rfiamcool/backoff/backoff_test.go
rfiamcool/backoff/backoff.go
rfiamcool/backoff/README.md
rfiamcool/grpcall
rfiamcool/grpcall/Gopkg.toml
rfiamcool/grpcall/design.jpg
rfiamcool/grpcall/format.go
```

fzf 用法多种多样

```
~/go/src/github.com
$ kill chrome tab
47/318
> 501 482 1 0 五 03下午 ?? 77:24.98 /Applications/Google Chrome.app/Contents/
501 542 482 0 五 03下午 ?? 11:12.63 /Applications/Google Chrome.app/Contents/
501 2270 482 0 五 03下午 ?? 2:26.89 /Applications/Google Chrome.app/Contents/
501 531 1 0 五 03下午 ?? 0:00.17 /Applications/Google Chrome.app/Contents/
501 25704 482 0 六 08下午 ?? 1:17.38 /Applications/Google Chrome.app/Contents/
501 19252 482 0 9:54下午 ?? 1:13.87 /Applications/Google Chrome.app/Contents/
501 20102 482 0 10:34下午 ?? 0:00.95 /Applications/Google Chrome.app/Contents/
501 22778 482 0 11:05下午 ?? 0:08.65 /Applications/Google Chrome.app/Contents/
```

ripgrep vs ag

~/go/src/github.com

```
$ time rg .vimrc|wc -l  
15
```

```
rg .vimrc 4.68s user 9.37s system 129% cpu 10.833 total  
wc -l 0.00s user 0.00s system 0% cpu 10.832 total
```

~/go/src/github.com

```
$ time ag .vimrc|wc -l  
15
```

```
ag .vimrc 2.97s user 8.57s system 130% cpu 8.867 total  
wc -l 0.00s user 0.00s system 0% cpu 8.866 total
```

~/go/src/github.com

```
$ time rg -n -w '[A-Z]+_SUSPEND'|wc -l  
133
```

```
rg -n -w '[A-Z]+_SUSPEND' 4.55s user 7.88s system 145% cpu 8.518 total  
wc -l 0.00s user 0.00s system 0% cpu 8.517 total
```

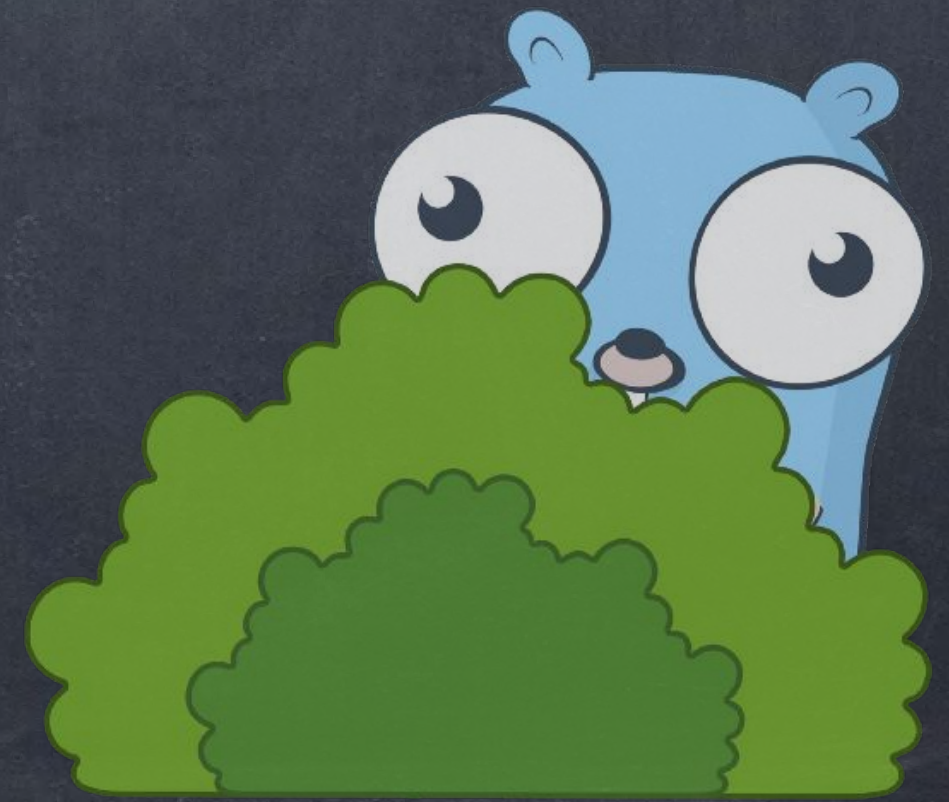
~/go/src/github.com

```
$ time ag -w '[A-Z]+_SUSPEND'|wc -l  
133
```

```
ag -w '[A-Z]+_SUSPEND' 3.93s user 6.70s system 161% cpu 6.568 total  
wc -l 0.00s user 0.00s system 0% cpu 6.567 total
```

• simple test

• ag > ripgrep > gnu grep



tldr

```
$ tldr nginx

nginx

Nginx web server.
More information: <https://nginx.org/en/>.

- Start server with the default config file:
  nginx

- Start server with a custom config file:
  nginx -c config_file

- Start server with a prefix for all relative paths in the config file:
  nginx -c config_file -p prefix/for/relative/paths

- Test the configuration without affecting the running server:
  nginx -t

- Reload the configuration by sending a signal with no downtime:
  nginx -s reload
```

```
$ tldr curl

curl

Transfers data from or to a server.
Supports most protocols, including HTTP, FTP, and POP3.
More information: <https://curl.haxx.se>.

- Download the contents of an URL to a file:
  curl http://example.com -o filename

- Download a file, saving the output under the filename indicated by the URL:
  curl -O http://example.com/filename

- Download a file, following [L]ocation redirects, and automatically [C]ontinuing (resuming) a previous download:
  curl -O -L -C - http://example.com/filename

- Send form-encoded data (POST request of type `application/x-www-form-urlencoded`). Use `--data` or `--data-raw` to specify the data:
  curl -d 'name=bob' http://example.com/form

- Send a request with an extra header, using a custom HTTP method:
  curl -H 'X-My-Header: 123' -X PUT http://example.com

- Send data in JSON format, specifying the appropriate content-type header:
  curl -d '{"name":"bob"}' -H 'Content-Type: application/json' http://example.com/users/1234

- Pass a user name and password for server authentication:
  curl -u myusername:mypassword http://example.com

- Pass client certificate and key for a resource, skipping certificate validation:
  curl --cert client.pem --key key.pem --insecure https://example.com
```

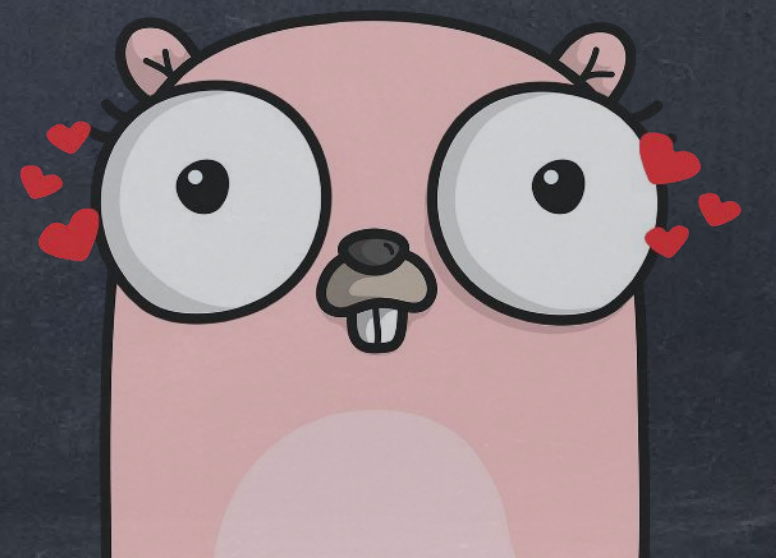
- 命令手册
- TL;DR
 - Too long, Don't read
- 比 man 更直观
- 比 man 有颜色
- 比 man 有常用的例子
- ...

fd

```
~/go/src/github.com/rfyiamcool/grpcall  🐱  master  ?  
$ fd 'en..r'  
enter.go  
  
~/go/src/github.com/rfyiamcool/grpcall  🐱  master  ?  
$ fd -e md  
README.md  
  
~/go/src/github.com/rfyiamcool/grpcall  🐱  master  ?  
$ fd -HI 'ser.*go$'  
testing/dynamic_register/server.go  
testing/test_server.go  
  
~/go/src/github.com/rfyiamcool/grpcall  🐱  master  ?  
$ fd -t f -x wc -l  
46 Gopkg.toml  
35 design.jpg  
124 format.go  
289 desc_source.go
```

vs find

- 👁 高亮
- 👁 更快
- 👁 更优雅
- 👁 并发管道



bat vs ccat

bat

- 丰富的颜色主题
- 自动分页
- 暗色行号
- more...

ccat

- 简单高亮
- 支持无后缀

```
File: ee.go
1 package main
2
3 import (
4     "fmt"
5     "net/http"
6     "time"
7 )
8
9 func hello(w http.ResponseWriter, req *http.Request) {
10
11     // A `context.Context` is created for each request by
12     // the `net/http` machinery, and is available with
13     // the `Context()` method.
14     ctx := req.Context()
15     fmt.Println("server: hello handler started")
16     defer fmt.Println("server: hello handler ended")
17
18     // Wait for a few seconds before sending a reply to the
19     // client. This could simulate some work the server is
20     // doing. While working, keep an eye on the context's
21     // `Done()` channel for a signal that we should cancel
22     // the work and return as soon as possible.
23     select {
24     case <-time.After(10 * time.Second):
25         fmt.Fprintf(w, "hello\n")
26     case <-ctx.Done():
27         // The context's `Err()` method returns an error
28         // that explains why the `Done()` channel was
29         // closed.
30         err := ctx.Err()
31         fmt.Println("server:", err)
32         internalError := http.StatusInternalServerError
33         http.Error(w, err.Error(), internalError)
34     }
```

```
File: ee.go
1 package main
2
3 import (
4     "fmt"
5     "net/http"
6     "time"
7 )
8
9 func hello(w http.ResponseWriter, req *http.Request) {
10
11     // A `context.Context` is created for each request by
12     // the `net/http` machinery, and is available with
13     // the `Context()` method.
14     ctx := req.Context()
15     fmt.Println("server: hello handler started")
16     defer fmt.Println("server: hello handler ended")
17
18     // Wait for a few seconds before sending a reply to the
19     // client. This could simulate some work the server is
20     // doing. While working, keep an eye on the context's
21     // `Done()` channel for a signal that we should cancel
22     // the work and return as soon as possible.
23     select {
24     case <-time.After(10 * time.Second):
25         fmt.Fprintf(w, "hello\n")
26     case <-ctx.Done():
27         // The context's `Err()` method returns an error
28         // that explains why the `Done()` channel was
29         // closed.
30         err := ctx.Err()
31         fmt.Println("server:", err)
32         internalError := http.StatusInternalServerError
```

diff

```
1 # shark
2
3 容器云管理系统
4
5 * k8s管理
6 * 机器初始化
7 * 监控采集 (node-exporter)
8 * hke管理
9
10 +-- 5 lines: ## 架构图
11
12 * hke-master
13 主控节点, 派发指令
14
15 * hke-syndic
16 中转节点, 缓解master的压力
17
18 * hke-minion
19 客户端, 执行指令及上报信息
20
21 ### Question
22
23
24
25
26
27
28
29
30
```

```
1 # shark
2
3 集群管理
4
5 ### 容器云管理系统
6
7 * k8s管理
8 * 机器初始化
9 * 监控采集 (node-exporter)
10 * hke管理
11
12 +-- 5 lines: # 架构图
13
14 * hke-master
15 主控节点, 派发指令
16
17 * hke-syndic
18 中转节点, 缓解master的压力
19
20 * rkhke-minion
21 客户端, 执行指令及上报信息
22
23 ### Question
24
25
26
27
28
29
30
```

```
@ master/master_server.go:6 @ package master
import (
    "context"
    "io"
    "io/ioutil"
    "net"
    "os"
    "path"
    "strings"
    "sync"
)

@ master/master_server.go:274 @ func (s *Server) GetStableMinionVersion(ctx context.Context, req *proto.MinionVe

// GetStableMinionBinary
func (s *Server) GetStableMinionBinary(ctx context.Context, req *proto.MinionVersion) (*proto.MinionBinary, error) {
    if conf.GetConfig().Master.StableMinionVersion == req.Version {
        if upgrader.version.String() == req.Version {
            return nil, grace.ErrSameVersion
        }
    }

    data, err := ioutil.ReadFile(path.Join(conf.MinionUpgradeReleasePath, req.Version))
    if err != nil {
        if os.IsNotExist(err) {
            return nil, err
        }
        log.Errorf("get minion upgrade release list failed, err: %v", err)
    }
    // limit minion's upgrade concurrently
    if err := upgrader.setLimiter(); err != nil {
        return nil, err
    }
}
```

- vimdiff
- icddiff
- diff-so-fancy for git

vim plug

```
美化主题
Plug 'Yggdrroot/indentLine'      " 美化代码缩进
Plug 'morhetz/gruvbox'          " colorscheme: gruvbox
Plug 'junegunn/seoul256.vim'    " colorscheme: seoul256
Plug 'w0ng/vim-hybrid'         " colorscheme: hybrid
Plug 'joshdick/onedark.vim'     " colorschema: onedark
Plug 'srcery-colors/srcery-vim' " colorschema: srcery
Plug 'rkr/vim-one'              " colorscheme: one
" Plug 'altercation/vim-colors-solarized' " 需要手动安装
Plug 'jaxbot/semantic-highlight.vim' " ,h => 根据语义来代码高亮
Plug 'bling/vim-airline'        " 状态栏
Plug 'vim-airline/vim-airline-themes' " 同上
```

```
34 Plug 'terryma/vim-expand-region' " + 扩大选择, - 减少选择
35 Plug 'ntpeters/vim-better-whitespace' " 空白
36 Plug 'fullybaked/toggle-numbers.vim' " 行号
37 Plug 'airblade/vim-gitgutter' " git状态
38 Plug 'lfv89/vim-interestingwords' " ,k ,K => 高亮操作的字符串
39 Plug 'elzr/vim-json' " json格式化
40 Plug 'ervandew/supertab' " 补全
41 Plug 'tpope/vim-surround' " 高效操作配对符号
42 Plug 'tpope/vim-repeat' " . => 重复surround操作
43 Plug 'jszakmeister/vim-togglecursor'
44 Plug 'junegunn/fzf', { 'dir': '~/.fzf', 'do': './install --all' }
45 Plug 'junegunn/fzf.vim' " 搜索
46 Plug 'mhinz/vim-startify' " vim启动菜单
47 Plug 'majutsushi/tagbar' " ,t => 列出tag
48 Plug 'tpope/vim-fugitive' " git命令
49 Plug 'junegunn/gv.vim' " git https://github.com/junegunn/gv.vim git commit br
50 Plug 'Xuyuanp/nerdtree-git-plugin' " 导航目录中看到 git 版本信息
51 Plug 'godlygeek/tabular' " 文本对齐, 选定文本 :Tab/符号
52 Plug 'jiangmiao/auto-pairs' " 自动匹配对称标记
53 Plug 'kien/rainbow_parentheses.vim' " 彩色括号
54 Plug 'honza/vim-snippets' " 代码模板
55 Plug 'SirVer/ultisnips' " 代码模板
56 Plug 'vim-syntastic/syntastic' " 语法检测
57 Plug 'scrooloose/nerdtree' " ,v => 目录树
58 Plug 'jistr/vim-nerdtree-tabs' " 目录树
59 Plug 'ctrlpvim/ctrlp.vim' " 类似vscode command + p
60 Plug 'tpope/vim-dispatch' " 异步编译及testing
61 Plug 'easymotion/vim-easymotion' " 快速移动插件, ',,w'可以在单词上出现前缀, 然后快速跳转
62 Plug 'haya14busa/incsearch.vim' " 支持regex的匹配查询
63 Plug 'scrooloose/nerdcommenter' " 快速注释
64 Plug 'haya14busa/incsearch-fuzzy.vim' " 同上
65 Plug 'Valloric/YouCompleteMe'
66 Plug 'dgryski/vim-godef' " for go lang
67 Plug 'fatih/vim-go', { 'do': ':GoInstallBinaries' }
68 Plug 'zchee/deoplete-iedl' " for python
```

easy-motion

```
16 → roundCount = 20000
17 )
18
19 func main() {
20     d
21     gddr := hlag.ktring("laddr", "q27.w.e.r:t0051", "ynput uerver iddr")
22     olag.parse()
23     z
24     xall(*cddr)
25 }
26 v
27 func hall(mddr ftring) {
28     jonn, ;ar := ;spc.;dal(
29     → ;gdr,
30     → ;hpc.;kthInsecure(),
31     )
32     ;l ;qr != ;wl {
33     → ;eg.;rtalf(";td ;yt ;unnect: %;i, ;or)
34     }
35     ;pfer ;znn.;xose()
36 ;c
NORMAL ↵ master# utf-8[unix] > +0 ~0 -0 > ~/go/src/github.com/rfyiamcool/grpcall/testing/dynamic_regi
Target key: █
```

- go to jump
- <leader><leader>s
- <leader><leader>b
- <leader><leader>w



fzf.vim

```
28
29 type jsonRequestParser struct {
30     dec          *json.Decoder
31     unmarshaler jsonpb.Unmarshaler
32 }
33
34 // NewJSONRequestParser returns a RequestParser that reads data in JSON format
35 // from the given reader.
36 func NewJSONRequestParser(in io.Reader, resolver jsonpb.AnyResolver) RequestParser {
COMMAND +0 ~0 -0 ! master format.go
invoke.go:213:4:// invokeAllStrem server and client are stream mode
invoke.go:195:20:      resp, err := stub.InvokeRpc(ctx, md, req, grpc.Trailer(&respTrailers), grpc.Header(&respHeader))
invoke.go:172:11:func (in *InvokeHandler) invokeUnary(ctx context.Context, stub grpcdynamic.Stub, md *desc.MessageDescriptor) (interface{}, error) {
invoke.go:170:12://      return invokeClientStream(ctx, stub, mtd, in.eventHandler, requestData, req)
invoke.go:165:13:      return in.invokeUnary(ctx, stub, mtd, in.eventHandler, requestData, req)
invoke.go:162:13:      return in.invokeServerStream(ctx, stub, mtd, in.eventHandler, requestData, req)
invoke.go:143:13:      return in.invokeAllStrem(ctx, stub, mtd, in.eventHandler, requestData, req, desc.MessageDescriptor())
invoke.go:83:11:func (in *InvokeHandler) InvokeRPC(ctx context.Context, source DescriptorSource, ch grpcdynamic.Channel) (interface{}, error) {
invoke.go:82:4:// InvokeRPC uses the given gRPC channel to invoke the given method.
invoke.go:76:10:      return &InvokeHandler{
invoke.go:75:9:func newInvokeHandler(event *EventHandler, inEvent InvocationEventHandler) *InvokeHandler {
invoke.go:70:6:type InvokeHandler struct {
README.md:110:18:a protoset file, invoke `protoc` with the `*.proto` files that define the service:
38/38
```

• command mode

• Files:

• GFiles:

• Rg: (ripgrep)

• Ag:

• History:

• ...



other vim

- [scroolouse/nerdcommenter](#)

- 快速注释

- [haya14busa/incsearch.vim](#)

- 支持正则的匹配 / 搜索

- [tpope/vim-surround](#)

- 高效引号标记操作

- [lfv89/vim-interestingwords](#)

- 单词高亮标记

- [brooth/far.vim](#)

- 字符串替换

- 宏

- 录制重复操作

- 文本对象

- 更快的匹配

- 寄存器

- 多个临时存储

- map & noremap

- 键盘映射

- 标记

- 自定义vimscript



do you know ?

- `fmt.Println("golang vs rust")`

- normal mode

- `f` `ang`

- `%`

- `w`

- `5l`

- `s`

- `$`

- `fmt.Println("golang vs rust")`

- normal mode

- `vi`

- `va`

- `di`

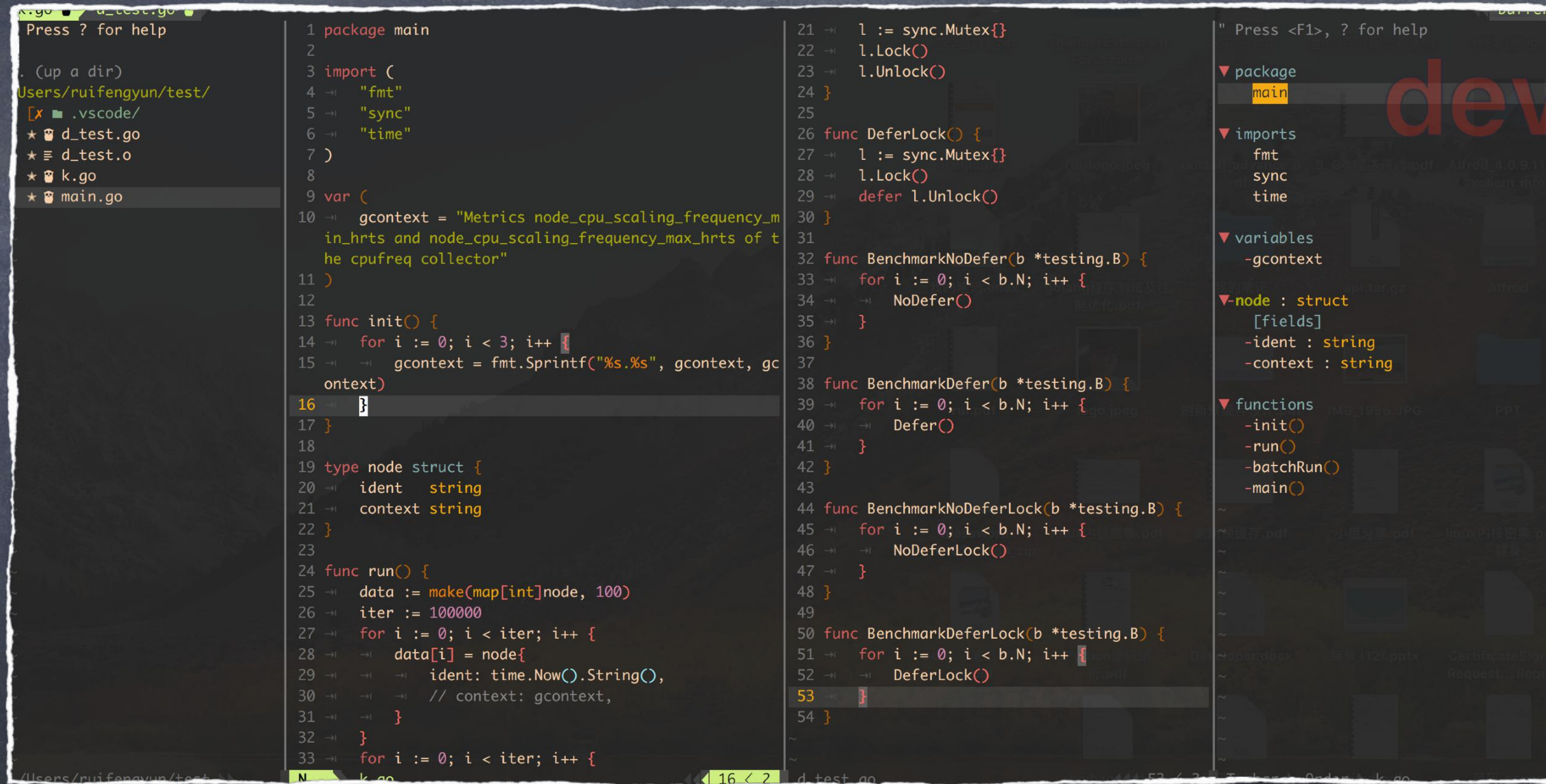
- `de`



如回答不出来, 说明你的vim还要多练习 !



vim config



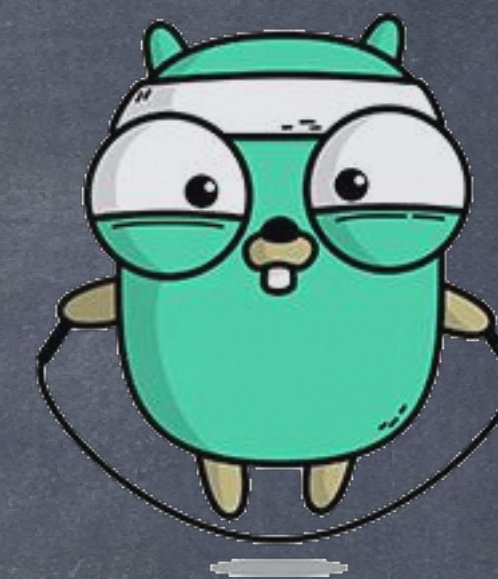
The screenshot displays the Vim editor interface with a Go program being edited. The left sidebar shows the file explorer with files like `d_test.go`, `d_test.o`, `k.go`, and `main.go`. The main editor area shows the following Go code:

```
1 package main
2
3 import (
4     "fmt"
5     "sync"
6     "time"
7 )
8
9 var (
10     gcontext = "Metrics node_cpu_scaling_frequency_max_hrts of the cpufreq collector"
11 )
12
13 func init() {
14     for i := 0; i < 3; i++ {
15         gcontext = fmt.Sprintf("%s.%s", gcontext, gcontext)
16     }
17 }
18
19 type node struct {
20     ident string
21     context string
22 }
23
24 func run() {
25     data := make(map[int]node, 100)
26     iter := 100000
27     for i := 0; i < iter; i++ {
28         data[i] = node{
29             ident: time.Now().String(),
30             // context: gcontext,
31         }
32     }
33     for i := 0; i < iter; i++ {
```

The right sidebar shows the documentation for the `package main`, including imports (`fmt`, `sync`, `time`), variables (`-gcontext`), and functions (`-init()`, `-run()`, `-batchRun()`, `-main()`).

https://github.com/rfyiamcool/dev_config





强烈建议
在常用的IDE里装入vim插件



多屏幕焦点切换



- 同一个应用不同显示器跳转
 - command + `
- 不同显示器之间的焦点跳转
 - mac自动操作 + cliclick
- alfred + cliclick 切换更多方位焦点



<https://github.com/rfyiamcool/multi-screen-jump>

tmux

split session

split windows

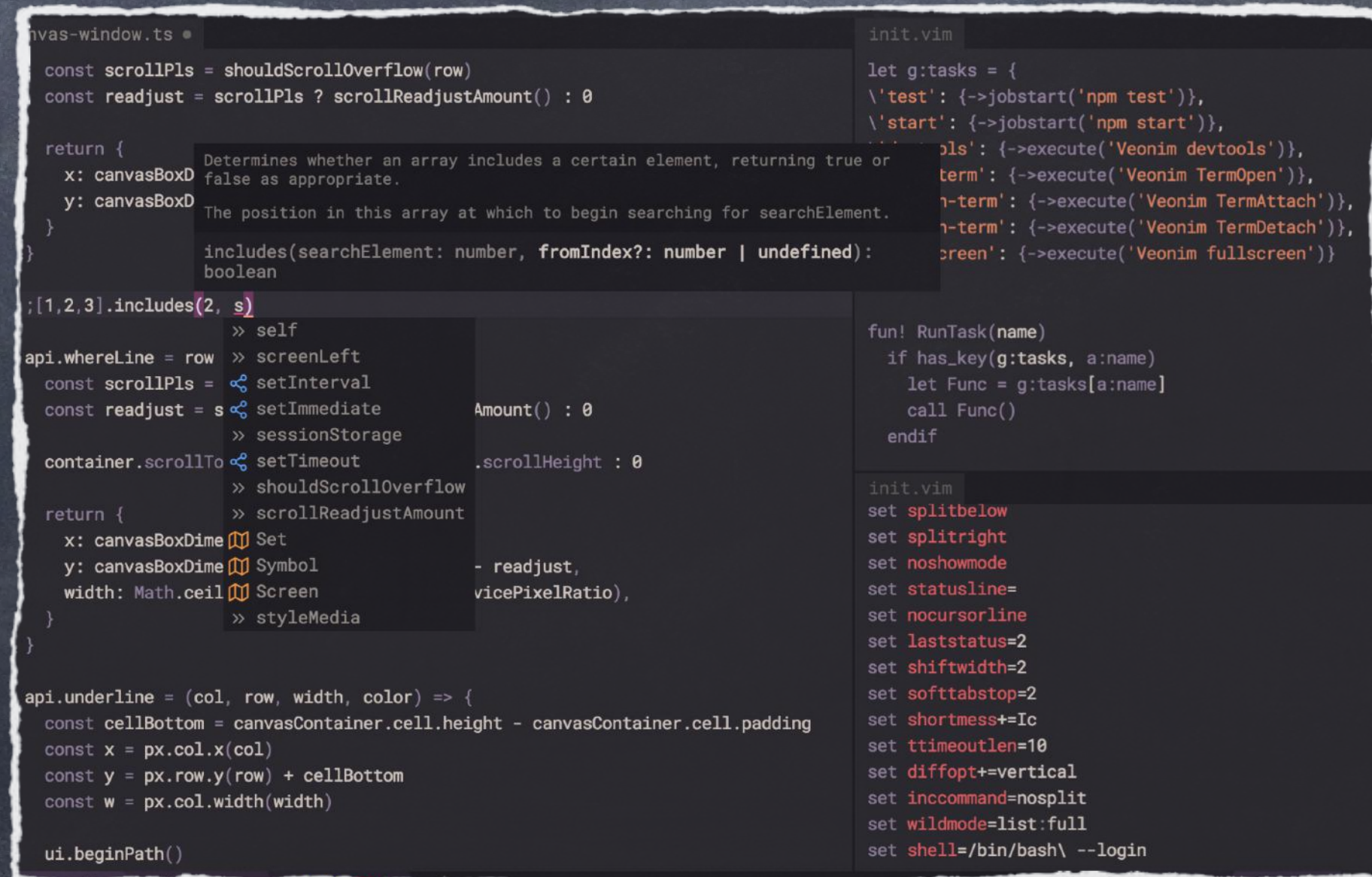
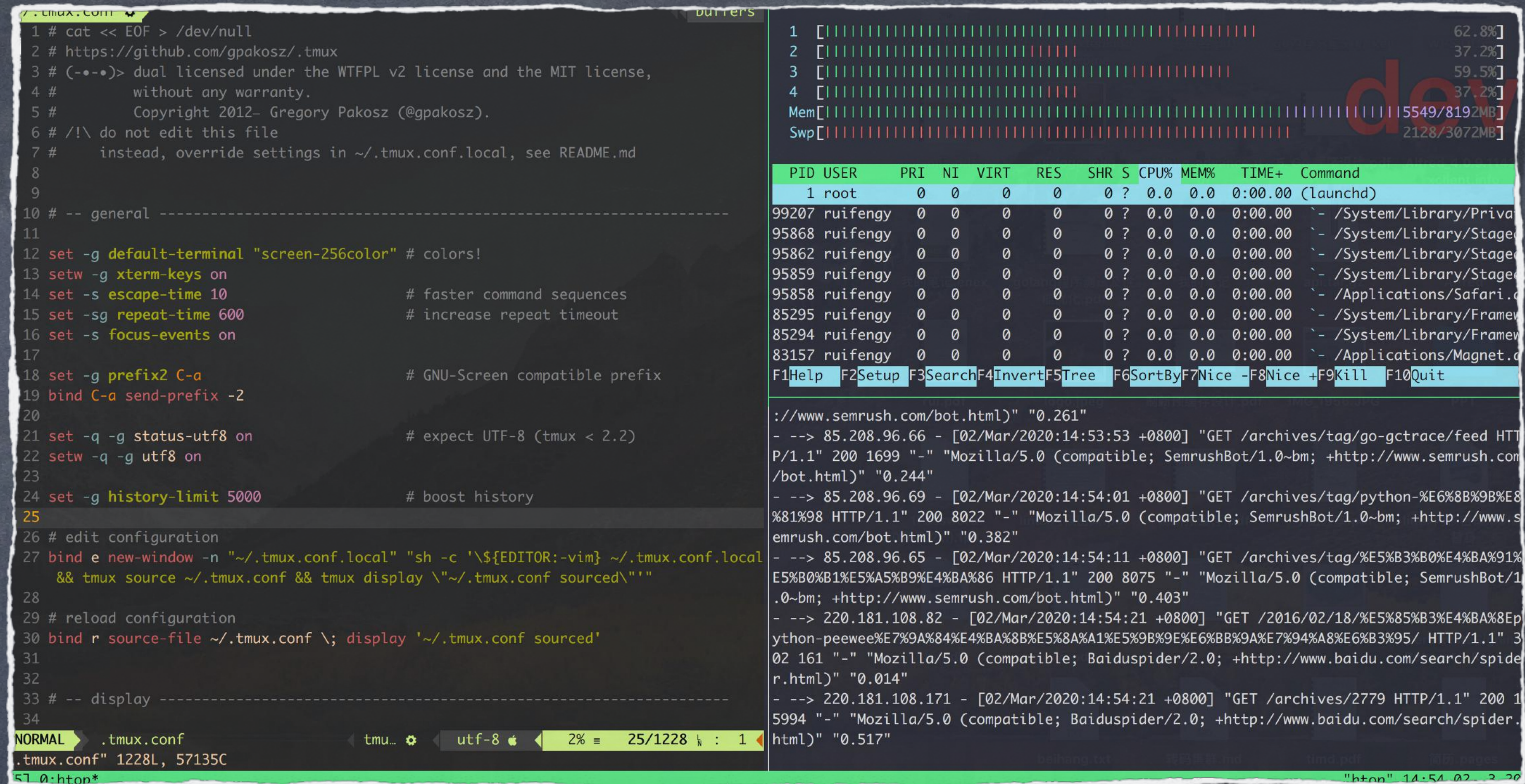
tmuxp

一键启动组合的tmux

neovim

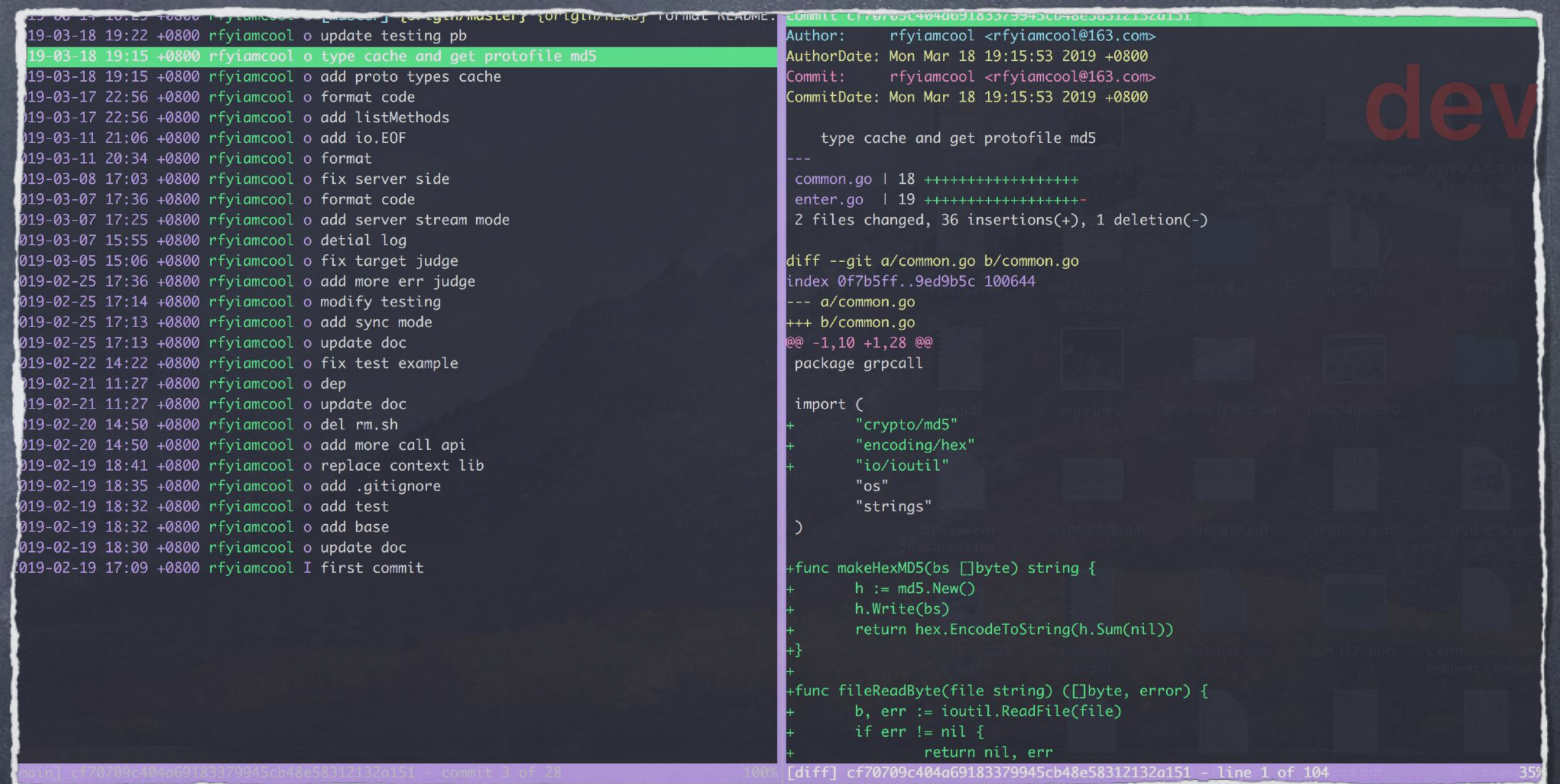
veonim

oni



other bin

- tig
- 更直观的git浏览工具
- pbcopy && pbpaste
- 对接mac的粘贴板
- tmuxp
- vscode -> code



```
19-03-18 19:22 +0800 rfyamcool o update testing pb
19-03-18 19:15 +0800 rfyamcool o type cache and get protofile md5
19-03-18 19:15 +0800 rfyamcool o add proto types cache
19-03-17 22:56 +0800 rfyamcool o format code
19-03-17 22:56 +0800 rfyamcool o add listMethods
19-03-11 21:06 +0800 rfyamcool o add io.EOF
19-03-11 20:34 +0800 rfyamcool o format
19-03-08 17:03 +0800 rfyamcool o fix server side
19-03-07 17:36 +0800 rfyamcool o format code
19-03-07 17:25 +0800 rfyamcool o add server stream mode
19-03-07 15:55 +0800 rfyamcool o detail log
19-03-05 15:06 +0800 rfyamcool o fix target judge
19-02-25 17:36 +0800 rfyamcool o add more err judge
19-02-25 17:14 +0800 rfyamcool o modify testing
19-02-25 17:13 +0800 rfyamcool o add sync mode
19-02-25 17:13 +0800 rfyamcool o update doc
19-02-22 14:22 +0800 rfyamcool o fix test example
19-02-21 11:27 +0800 rfyamcool o dep
19-02-21 11:27 +0800 rfyamcool o update doc
19-02-20 14:50 +0800 rfyamcool o del rm.sh
19-02-20 14:50 +0800 rfyamcool o add more call api
19-02-19 18:41 +0800 rfyamcool o replace context lib
19-02-19 18:35 +0800 rfyamcool o add .gitignore
19-02-19 18:32 +0800 rfyamcool o add test
19-02-19 18:32 +0800 rfyamcool o add base
19-02-19 18:30 +0800 rfyamcool o update doc
19-02-19 17:09 +0800 rfyamcool I first commit

Commit: cf70709c404a69183379945cb48e58312132a151
Author: rfyamcool <rfyamcool@163.com>
AuthorDate: Mon Mar 18 19:15:53 2019 +0800
Commit: rfyamcool <rfyamcool@163.com>
CommitDate: Mon Mar 18 19:15:53 2019 +0800

type cache and get protofile md5
---
common.go | 18 +++++
enter.go | 19 +++++
2 files changed, 36 insertions(+), 1 deletion(-)

diff --git a/common.go b/common.go
index 0f7b5ff..9ed9b5c 100644
--- a/common.go
+++ b/common.go
@@ -1,10 +1,28 @@
package grpcall

import (
+    "crypto/md5"
+    "encoding/hex"
+    "io/ioutil"
+    "os"
+    "strings"
)

+func makeHexMD5(bs []byte) string {
+    h := md5.New()
+    h.Write(bs)
+    return hex.EncodeToString(h.Sum(nil))
+}

+func fileReadByte(file string) ([]byte, error) {
+    b, err := ioutil.ReadFile(file)
+    if err != nil {
+        return nil, err
+    }
+}
```

develop

- expect 自动化脚本
 - 校验 google 认证
 - ssh 登陆目标主机
- fswatch-run
 - 监听目录变化
 - 自动测试
 - 自动重启服务
 - 实时推送代码到开发服务器



```
...

function sync() {
    for host in $hosts
    do
        rsync -avz ${delete} --rsh="ssh -p${port}" --progress --exclude="*.log" ${exclude} ${src}
    root@${host}:${dst}
    done
}

function execute() {
    for host in $hosts
    do
        ssh -o ConnectTimeout=1 -l root ${host} $1
        if [ $? -ne 0 ];then
            exit 99
        fi
    done
}

function watch() {
    fswatch -l 1 -0 $src |while read -d "" event; do

        now=`get_time`

        echo "will sync " ${event} ${ts} ${now}
        if [[ ${current} == ${now} ]]; then
            echo "filter...."
            continue
        fi
        current=`get_time`

        sync

    done
}

...
```

develop

- alias 各种命令别名
- 自定义 snippet 模板
- 请尽量多记使用软件的快捷键
 - 各种快捷键
 - 各种快捷键
 - 各种快捷键

```
42     "tmpl-workermodel": {
43         "prefix": "tmpl-workermodel",
44         "body": [
45             "type $1 struct {}\\n",
46             "func New$1 () *$1 {\\n return &$1{} }\\n",
47             "func (w *$1 ) Start() {\\n}\\n",
48             "func (w *$1 ) Stop() {\\n}\\n",
49             "func (w *$1 ) Restart() {\\n}\\n",
50             "func (w *$1 ) Reload() {\\n}\\n",
51             "func (w *$1 ) SetConfig() {\\n}\\n",
52         ],
53     },
54     "tmpl-ctx": {
55         "prefix": "tmpl-ctx",
56         "body": [
57             "ctx, cancel := context.WithCancel(context.Background())"
58         ],
59     },
60     "tmpl-select": {
61         "prefix": "tmpl-select",
62         "body": [
63             "select {",
64             "\\tcase <- $1:",
65             "\\t\tf, _ := $1()",
```



设备篇



- 键盘
- 桌子
- 椅子
- 显示器
- ...

键盘

👁️ 机械键盘

👁️ leopold 👍

👁️ filco 😅

👁️ ikbc 😊

👁️ keychron 😭

👁️ 静电容

👁️ hhkb 👍

👁️ leopold 😊

👁️ realforce 🤔

建议 红轴

在办公室听到青轴声音后，莫名伤感 ...



键盘



椅子



西昊 m57

¥ 1000.00

冈村日本okamura

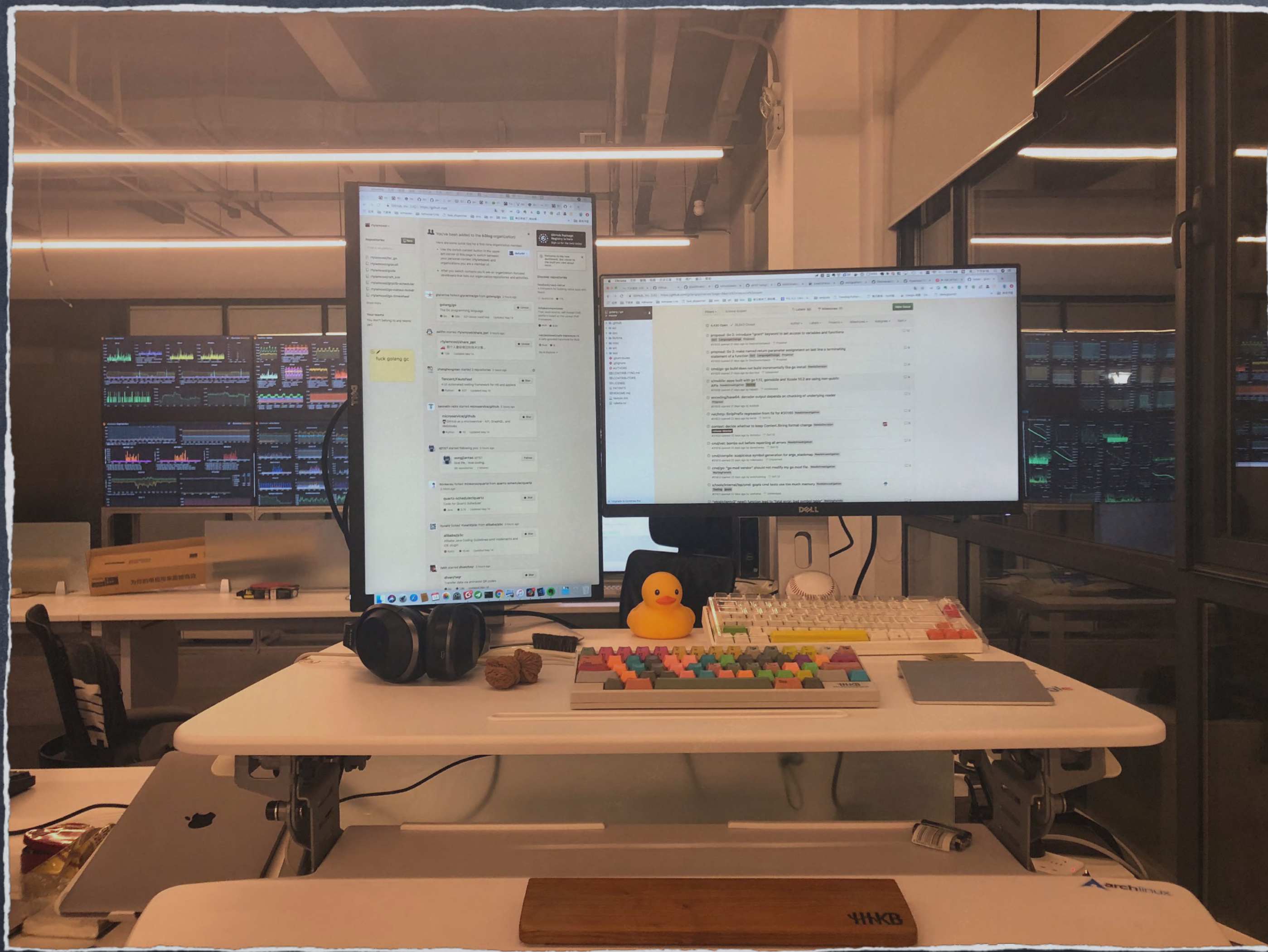
sylphy

¥ 4000.00

contessa

¥ 11000.00

Herman Miller



👁️ 对腰&前列腺

👁️ 可拔高的桌子 (乐歌 m1m)

👁️ 对眼

👁️ 暖色及暗色主题

👁️ 对手

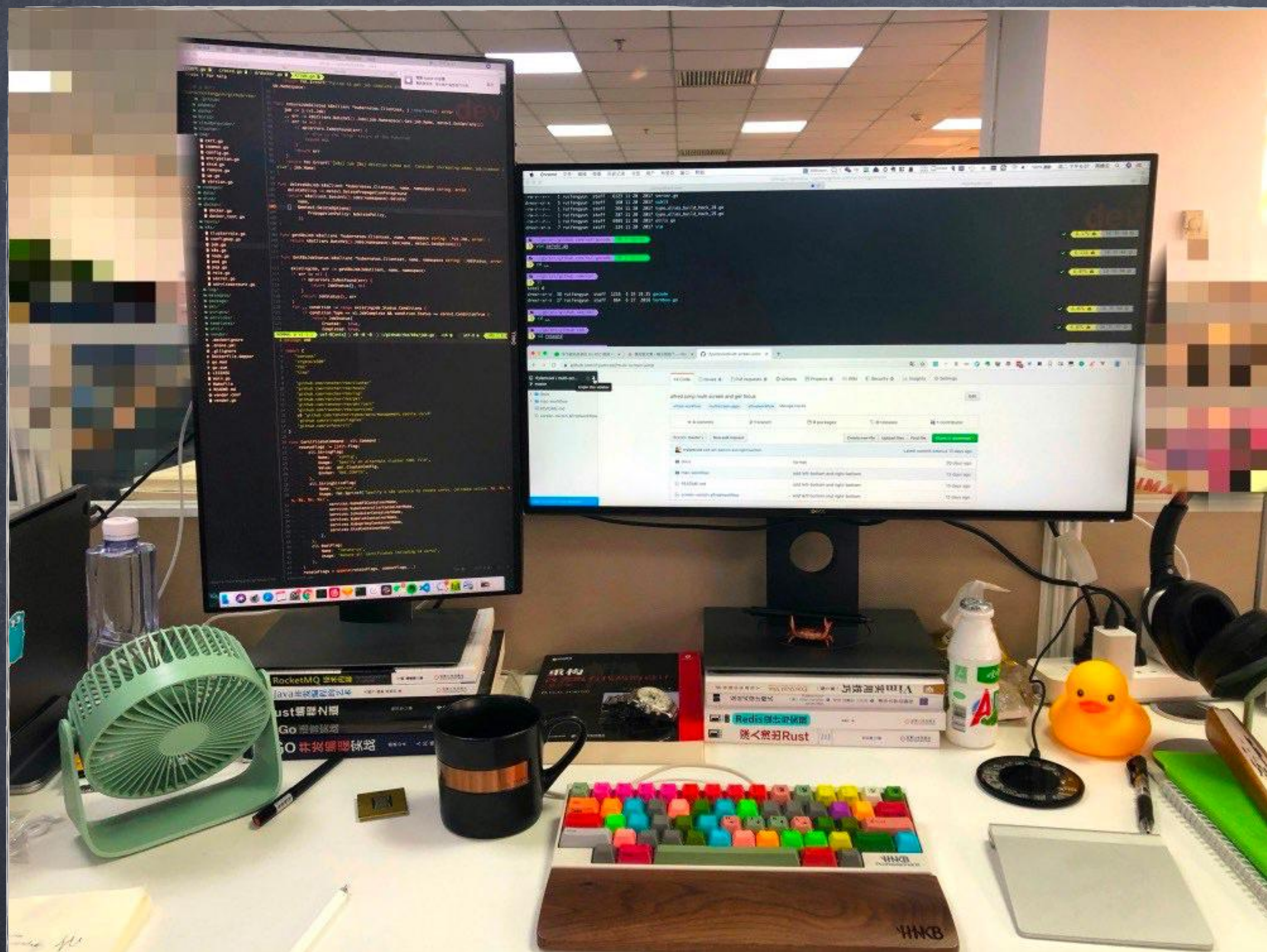
👁️ 核桃

👁️ 棒球

👁️ 对bug

👁️ 按鸭子

显示器



一横屏（辅助）

一竖屏（主屏）

强烈建议

无需大角度摇头头部

- 👁 4k
- 👁 LG 4k 27UL650
- 👁 DELL U2720Q
- 👁 Benq PD2700U
- 👁 more ...



终结篇:



- 在桌子上需摆放
 - 深入理解linux内核
 - tcp/ip 详解
 - unix环境高级编程
 - 图解电动车维修
 - ...

“ Q&A ”

– xiaorui.cc

– github.com/rfyiamcool

