

虚拟机的使用 (ubuntu20.04)



切换软件源

进入software-->"Download from:"中选择"other"-->"Select Bset Sever"-
->"choose Sever"-->"close"-->"Reload"

切换系统语言为中文

"settings" --> "Region & Language" --> "Manage Installed Languages" -
-> "install/Remove Language" --> 勾选"chinese(simplified)" --> "Apply"
--> 将"Language"下的"汉语(中国)"拖拽到前三行 --> "Apply System-
Wide" --> 将系统注销,并重新登录

搜狗输入法的安装

参考网址: <https://shurufa.sogou.com/linux/guide>

VIM的安装和配置

安装

输入命令安装vim: `sudo apt install vim`

配置

原文链接: https://blog.csdn.net/qq_42007287/article/details/124687587

下面是网上荡的vimrc文件，功能基本完善，可以直接完全复制到你的vimrc文件中。如果你想定制你专属的vim编辑器，把文件内容中的Author字段改成你的昵称，Mail字段后面改成你的邮箱号，这样你就拥有你的专属vim编辑器了。

Author: username

Mail: ☁ 111111111@qq.com

```
1 set sw=4
2 set ts=4
3 set et
4 set smarttab
5 set smartindent
6 set lbr
7 set fo+=mB
8 set sm
9 set selection=inclusive
10 set wildmenu
11 set mousemodel=popup
12
13 set pastetoggle=<F11>
14
15 au FileType php setlocal
    dict+=~/.vim/dict/php_funclist.dict
16 au FileType css setlocal dict+=~/.vim/dict/css.dict
17 au FileType c setlocal dict+=~/.vim/dict/c.dict
18 au FileType cpp setlocal dict+=~/.vim/dict/cpp.dict
19 au FileType scale setlocal dict+=~/.vim/dict/scale.dict
20 au FileType javascript setlocal
    dict+=~/.vim/dict/javascript.dict
21 au FileType html setlocal
    dict+=~/.vim/dict/javascript.dict
22 au FileType html setlocal dict+=~/.vim/dict/css.dict
23
24 "
25 "syntastic相关
26 "execute pathogen#infect()
```

```

27 let g:syntastic_python_checkers=['pylint']
28 let g:syntastic_php_checkers=['php', 'phpcs', 'phpmd']
29 "golang
30 "Processing... % (ctrl+c to stop)
31 let g:fencview_autodetect=0
32 set rtp+=$GOROOT/misc/vim
33 "
34 " 显示相关
35 "
36 syntax on
37 set cul "高亮光标所在行
38 set cuc
39 set shortmess=atI " 启动的时候不显示那个援助乌干达儿童的提示
40 set go= " 不要图形按钮
41 "color desert " 设置背景主题
42 color ron " 设置背景主题
43 "color torte " 设置背景主题
44 "set guifont=Courier_New:h10:cANSI " 设置字体
45 "autocmd InsertLeave * se nocul " 用浅色高亮当前行
46 autocmd InsertEnter * se cul " 用浅色高亮当前行
47 set ruler " 显示标尺
48 set showcmd " 输入的命令显示出来，看的清楚些
49 "set whichwrap+=<,>,h,l " 允许backspace和光标键跨越行边界(不建议)
50 set scrolloff=3 " 光标移动到buffer的顶部和底部时保持3行距离
51 set statusline=%F%m%r%h%w\ [FORMAT=%{&ff}]\ [TYPE=%Y]\ [POS=%l,%v][%p%]\ %{strftime("\%d/%m/%y\ -\ %H:%M\" )}
    "状态行显示的内容
52 set laststatus=2 " 启动显示状态行(1),总是显示状态行(2)
53 "set foldenable " 允许折叠
54 ""set foldmethod=manual " 手动折叠
55 set nocompatible "去掉讨厌的有关vi一致性模式，避免以前版本的一些bug和局限
56 " 显示中文帮助
57 if version >= 603

```

```
58         set helplang=cn
59         set encoding=utf-8
60     endif
61     " 自动缩进
62     set autoindent
63     set cindent
64     " Tab键的宽度
65     set tabstop=4
66     " 统一缩进为4
67     set softtabstop=4
68     set shiftwidth=4
69     " 使用空格代替制表符
70     set expandtab
71     " 在行和段开始处使用制表符
72     set smarttab
73     " 显示行号
74     set number
75     " 历史记录数
76     set history=1000
77     "搜索逐字符高亮
78     set hlsearch
79     set incsearch
80     "语言设置
81     set langmenu=zh_CN.UTF-8
82     set helplang=cn
83     " 总是显示状态行
84     set cmdheight=2
85     " 侦测文件类型
86     filetype on
87     " 载入文件类型插件
88     filetype plugin on
89     " 为特定文件类型载入相关缩进文件
90     filetype indent on
91     " 保存全局变量
92     set viminfo+=!
93     " 带有如下符号的单词不要被换行分割
94     set iskeyword+=_,$,@,%,#,-
95     " 字符间插入的像素行数目
96
```

```

97  "markdown配置
98  au BufRead,BufNewFile *.
    {md,mdown,mkd,mkdn,markdown,mdwn}    set filetype=mkd
99  au BufRead,BufNewFile *.{go}    set filetype=go
100 au BufRead,BufNewFile *.{js}    set filetype=javascript
101 "rkdown to HTML
102 nmap md :!~/./vim/markdown.pl % > %.html <CR><CR>
103 nmap fi :!firefox %.html & <CR><CR>
104 nmap \ \cc
105 vmap \ \cc
106
107 "将tab替换为空格
108 nmap tt :%s/\t/      /g<CR>
109
110
111
112 "=====
113 "=====
113 "===="新文件标题
114 "=====
115 "=====
115 "新建.c,.h,.sh,.java文件, 自动插入文件头
116 autocmd BufNewFile *.cpp,*.ch,*.sh,*.rb,*.java,*.py
    exec ":call SetTitle()"
117 ""定义函数SetTitle, 自动插入文件头
118 func SetTitle()
119     "如果文件类型为.sh文件
120     if &filetype == 'sh'
121         call setline(1,"#!/bin/bash")
122         call append(line("."), "")
123     elseif &filetype == 'python'
124         call setline(1,"#!/usr/bin/env python")
125         call append(line("."), "# coding=utf-8")
126         call append(line(".")+1, "")
127
128     elseif &filetype == 'ruby'
129         call setline(1,"#!/usr/bin/env ruby")
130         call append(line("."), "# encoding: utf-8")
131         call append(line(".")+1, "")

```

```

132
133 "    elseif &filetype == 'mkd'
134 "        call setline(1,"<head><meta charset=\"UTF-8\">
</head>")
135     else
136         call setline(1,
"/*****
*****")
137         call append(line("."), "        > File
Name: ".expand("%"))
138         call append(line(".")+1, "        >
Author: username")
139         call append(line(".")+2, "        >
Mail: 11111111@qq.com")
140         call append(line(".")+3, "        >
Created Time: ".strftime("%c"))
141         call append(line(".")+4, "
*****
*****/")
142         call append(line(".")+5, "")
143     endif
144     if expand("%:e") == 'cpp'
145         call append(line(".")+6, "#include
<iostream>")
146         call append(line(".")+7, "using std::cin;")
147         call append(line(".")+8, "using std::cout;")
148         call append(line(".")+9, "using std::endl;")
149         call append(line(".")+10, "using
namespace std;")
150         call append(line(".")+11, "")
151     endif
152     if &filetype == 'c'
153         call append(line(".")+6, "#include
<stdio.h>")
154         call append(line(".")+7, "")
155     endif
156     if expand("%:e") == 'h'
157         call append(line(".")+6, "#ifndef
_".toupper(expand("%:r"))."_H")

```

```

158             call append(line(".")+7, "#define
_.toupper(expand("%:r"))."_H")
159             call append(line(".")+8, "#endif")
160         endif
161         if &filetype == 'java'
162             call append(line(".")+6,"public class
_.expand("%:r"))
163             call append(line(".")+7,"")
164         endif
165         "新建文件后，自动定位到文件末尾
166     endfunc
167 autocmd BufNewFile * normal G
168 command WQ wq
169 command Wq wq
170 "=====
171 "键盘命令
172 "=====
173 :nmap <silent> <F9> <ESC>:Tlist<RETURN>
174 " shift tab pages
175 map <S-Left> :tabp<CR>
176 map <S-Right> :tabn<CR>
177 map! <C-Z> <Esc>zzi
178 map! <C-O> <C-Y>,
179 map <C-A> ggVG$" +y
180 map <Esc><Esc> :w<CR>
181 map <F12> gg=G
182 map <C-w> <C-w>w
183 imap <C-k> <C-y>,
184 imap <C-t> <C-q><TAB>
185 imap <C-j> <ESC>
186 " 选中状态下 Ctrl+c 复制
187 "map <C-v> "*"pa
188 imap <C-v> <Esc>*"pa
189 imap <C-a> <Esc>^
190 imap <C-e> <Esc>$
191 vmap <C-c> " +y
192 "set mouse=v

```

```
193 "set clipboard=unnamed
194 "去注释
195 nnoremap <F1> :g/^\s*#/d<CR>
196 "去空行
197 nnoremap <F2> :g/^\s*$/d<CR>
198 "比较文件
199 nnoremap <C-F2> :vert diffsplit
200 "nnoremap <Leader>fu :CtrlPFunky<Cr>
201 "nnoremap <C-n> :CtrlPFunky<Cr>
202 "列出当前目录文件
203 map <F3> :NERDTreeToggle<CR>
204 imap <F3> <ESC> :NERDTreeToggle<CR>
205 "打开树状文件目录
206 map <C-F3> \be
207 :autocmd BufRead,BufNewFile *.dot map <F5> :w<CR>:!dot
    -Tjpg -o %<.jpg % && eog %<.jpg <CR><CR> && exec
    "redr!"
208 "C, C++ 按F5编译运行
209 map <F5> :call CompileRunGcc()<CR>
210 func! CompileRunGcc()
211     exec "w"
212     if &filetype == 'c'
213         exec "!g++ % -o %<"
214         exec "!time ./%<"
215     elseif &filetype == 'cpp'
216         exec "!g++ % -std=c++11 -o %<"
217         exec "!time ./%<"
218     elseif &filetype == 'java'
219         exec "!javac %"
220         exec "!time java %<"
221     elseif &filetype == 'sh'
222         :!time bash %
223     elseif &filetype == 'python'
224         exec "!time python2.7 %"
225     elseif &filetype == 'html'
226         exec "!firefox % &"
227     elseif &filetype == 'go'
228         "     exec "!go build %<"
229         exec "!time go run %"
```



```
230     elseif &filetype == 'mkd'
231         exec "!~/vim/markdown.pl % > %.html &"
232         exec "!firefox %.html &"
233     endif
234 endfunc
235 "C,C++的调试
236 map <F8> :call Rungdb()<CR>
237 func! Rungdb()
238     exec "w"
239     exec "!g++ % -std=c++11 -g -o %<"
240     exec "!gdb ./%<"
241 endfunc
242
243
244 "代码格式优化
245
246 map <F6> :call FormartSrc()<CR><CR>
247
248 "定义FormartSrc()
249 func FormartSrc()
250     exec "w"
251     if &filetype == 'c'
252         exec "!astyle --style=ansi -a --suffix=none %"
253     elseif &filetype == 'cpp' || &filetype == 'hpp'
254         exec "r !astyle --style=ansi --one-line=keep-
statements -a --suffix=none %> /dev/null 2>&1"
255     elseif &filetype == 'perl'
256         exec "!astyle --style=gnu --suffix=none %"
257     elseif &filetype == 'py' || &filetype == 'python'
258         exec "r !autopep8 -i --aggressive %"
259     elseif &filetype == 'java'
260         exec "!astyle --style=java --suffix=none %"
261     elseif &filetype == 'jsp'
262         exec "!astyle --style=gnu --suffix=none %"
263     elseif &filetype == 'xml'
264         exec "!astyle --style=gnu --suffix=none %"
265     else
266         exec "normal gg=G"
267     return
```

[illegible]

```
301 "set cursorline          " 突出显示当前行
302 set magic                " 设置魔术
303 set guioptions-=T        " 隐藏工具栏
304 set guioptions-=m        " 隐藏菜单栏
305 ""set foldcolumn=0
306 ""set foldmethod=indent
307 ""set foldlevel=3
308 " 不要使用vi的键盘模式，而是vim自己的
309 set nocompatible
310 " 去掉输入错误的提示声音
311 set noeb
312 " 在处理未保存或只读文件的时候，弹出确认
313 set confirm
314 "禁止生成临时文件
315 set nobackup
316 set noswapfile
317 "搜索忽略大小写
318 set ignorecase
319
320
321
322
323 set linespace=0
324 " 增强模式中的命令行自动完成操作
325 set wildmenu
326 " 使回格键（backspace）正常处理indent, eol, start等
327 set backspace=2
328 " 允许backspace和光标键跨越行边界
329 set whichwrap+=<,>,h,l
330 " 可以在buffer的任何地方使用鼠标（类似office中在工作区双击鼠标定位）
331 "set mouse=a
332 set selection=exclusive
333 set selectmode=mouse,key
334 " 通过使用： commands命令，告诉我们文件的哪一行被改变过
335 set report=0
336 " 在被分割的窗口间显示空白，便于阅读
337 set fillchars=vert:\ ,stl:\ ,stlnc:\
338 " 高亮显示匹配的括号
```

```

339 set showmatch
340 " 匹配括号高亮的时间（单位是十分之一秒）
341 set matchtime=1
342 " 光标移动到buffer的顶部和底部时保持3行距离
343 set scrolloff=3
344 " 为C程序提供自动缩进
345 "自动补全
346 "" :inoremap ( (<ESC>i
347 "" :inoremap ) <c-r>=ClosePair('')<CR>
348 "" :inoremap { {<CR><ESC>O
349 "" :inoremap } <c-r>=ClosePair('}')<CR>
350 "" :inoremap [ [<ESC>i
351 "" :inoremap ] <c-r>=ClosePair(']')<CR>
352 "" :inoremap " "<ESC>i
353 "" :inoremap ' '<ESC>i
354 ""function! ClosePair(char)
355 ""     if getline('.')[col('.') - 1] == a:char
356 ""         return "<Right>"
357 ""     else
358 ""         return a:char
359 ""     endif
360 ""endfunction
361 filetype plugin indent on
362 "打开文件类型检测，加了这句才可以用智能补全
363 set completeopt=longest,menu
364 ""
365 ""
366 ""
367 let Tlist_Sort_Type = "name"      " 按照名称排序
368 let Tlist_Use_Right_Window = 1    " 在右侧显示窗口
369 let Tlist_Compart_Format = 1      " 压缩方式
370 let Tlist_Exist_OnlyWindow = 1    " 如果只有一个buffer，
    kill窗口也kill掉buffer
371 ""let Tlist_File_Fold_Auto_Close = 0 " 不要关闭其他文件的
    tags
372 ""let Tlist_Enable_Fold_Column = 0    " 不要显示折叠树

```

```

373 "let Tlist_Show_One_File=1 "不同时显示多个文件
    的tag, 只显示当前文件的
374 "设置tags
375 set tags=tags;
376 "set autochdir
377
378
379 """"""""""
    """"""""""
    """"
380 "其他东东
381 """"""""""
    """"""""""
    """"
382 "默认打开Taglist
383 let Tlist_Auto_Open=0
384 """"""""""
385 " Tag list (ctags)
386 """"""""""
387 let Tlist_Ctags_Cmd = '/usr/local/bin/ctags'
388 let Tlist_Show_One_File = 1 "不同时显示多个文件的tag, 只显
    示当前文件的
389 let Tlist_File_Fold_Auto_Close = 1
390 let Tlist_Exit_OnlyWindow = 1 "如果taglist窗口是最后一个窗
    口, 则退出vim
391 let Tlist_Use_Right_Window = 1 "在右侧窗口中显示taglist窗
    口
392 " minibufexpl插件的一般设置
393 let g:miniBufExplMapWindowNavVim = 1
394 let g:miniBufExplMapWindowNavArrows = 1
395 let g:miniBufExplMapCTabSwitchBufs = 1
396 let g:miniBufExplModSelTarget = 1
397 nmap tl :Tlist<cr>
398
399 "python补全
400 let g:pydiction_location = '~/vim/after/complete-dict'
401 let g:pydiction_menu_height = 20
402 let Tlist_Ctags_Cmd='/usr/local/bin/ctags'
403 let g:miniBufExplMapWindowNavVim = 1

```

```
404 let g:miniBufExplMapWindowNavArrows = 1
405 let g:miniBufExplMapCTabSwitchBufs = 1
406 let g:miniBufExplModSelTarget = 1
407
408
409 set iskeyword+=.
410 set termencoding=utf-8
411 set encoding=utf8
412 set fileencodings=utf8,ucs-bom,gbk,cp936,gb2312,gb18030
413
414 autocmd FileType python set
    omnifunc=pythoncomplete#Complete
415
416 "set nocompatible          " be iMproved
417 "filetype off              " required!
418
419 set rtp+=~/.vim/bundle/vundle/
420 "call vundle#rc()
421
422 " let Vundle manage Vundle
423 " required!
424 "Bundle 'gmarik/vundle'
425
426 " My Bundles here:
427 "
428 " original repos on github
429 " Bundle 'tpope/vim-fugitive'
430 " Bundle 'rstacruz/sparkup', {'rtp': 'vim/'}
431 " Bundle 'Yggdrroot/indentLine'
432 let g:indentLine_char = '|'
433 "ndle 'tpope/vim-rails.git'
434 " vim-scripts repos
435 " Bundle 'L9'
436 " Bundle 'FuzzyFinder'
437 " non github repos
438 " Bundle 'https://github.com/w Vincent/command-t.git'
439 " Bundle 'Auto-Pairs'
440 " Bundle 'python-imports.vim'
441 " Bundle 'CaptureClipboard'
```

```
442 " Bundle 'ctrlp-modified.vim'
443 " Bundle 'last_edit_marker.vim'
444 " Bundle 'synmark.vim'
445 "Bundle 'Python-mode-klen'
446 " Bundle 'SQLComplete.vim'
447 " Bundle 'Javascript-OmniCompletion-with-YUI-and-j'
448 "Bundle 'JavaScript-Indent'
449 "Bundle 'Better-Javascript-Indentation'
450 "Bundle 'jslint.vim'
451 "Bundle "pangloss/vim-javascript"
452 "Bundle 'Vim-Script-Updater'
453 "Bundle 'ctrlp.vim'
454 "Bundle 'tacahiroy/ctrlp-funky'
455 "Bundle 'jsbeautify'
456 "Bundle 'The-NERD-Commenter'
457 "django
458 "Bundle 'django_templates.vim'
459 "Bundle 'Django-Projects'
460
461 "Bundle 'FredKSchott/CoVim'
462 "Bundle 'djangojump'
463 " ...
464 let g:html_indent_inctags = "html,body,head,tbody"
465 let g:html_indent_script1 = "inc"
466 let g:html_indent_style1 = "inc"
467
468 filetype plugin indent on      " required!
469 "
470 "ctrlp设置
471 "
472 set
  wildignore+=*/tmp/*,*.so,*.swp,*.zip,*.pyc,*.png,*.jpg,
  *.gif      " MacOSX/Linux
473 set
  wildignore+=*\\tmp\\*,*.swp,*.zip,*.exe,*.pyc,*.png,*.j
  pg,*.gif  " Windows
474
475 let g:ctrlp_custom_ignore = '\v[\/]\.(git|hg|svn)$'
476 let g:ctrlp_custom_ignore = '\v\.(exe|so|dll)$'
```

```
477 let g:ctrlp_extensions = ['funky']
478
479 let NERDTreeIgnore=['\\.pyc']
480
```

C/C++编译器的安装

首先必须是root用户,切换root用户使用下面的命令 然后输入root用户的密码

```
1 su root
```

Ubuntu 安装后, root用户默认是被锁定了的, 不允许登录, 也不允许 "su" 到 root。所以这里我遇到了这样的问题:

问题	解决方法
<pre>user-1@ubuntu:~\$ su root 密码: su: 认证失败</pre>	<pre>user-1@ubuntu:~\$ sudo passwd [sudo] user-1 的密码: 新的 密码: 重新输入新的 密码: passwd: 已成功更新密码 user-1@ubuntu:~\$ su root 密码: root@ubuntu:/home/user-1#</pre>

接下来, 就可以装C语言编译器gcc:

```
1 sudo apt update
2
3 //该命令会安装大量软件包, 包括gcc, g++
4 sudo apt install build-essential
```
