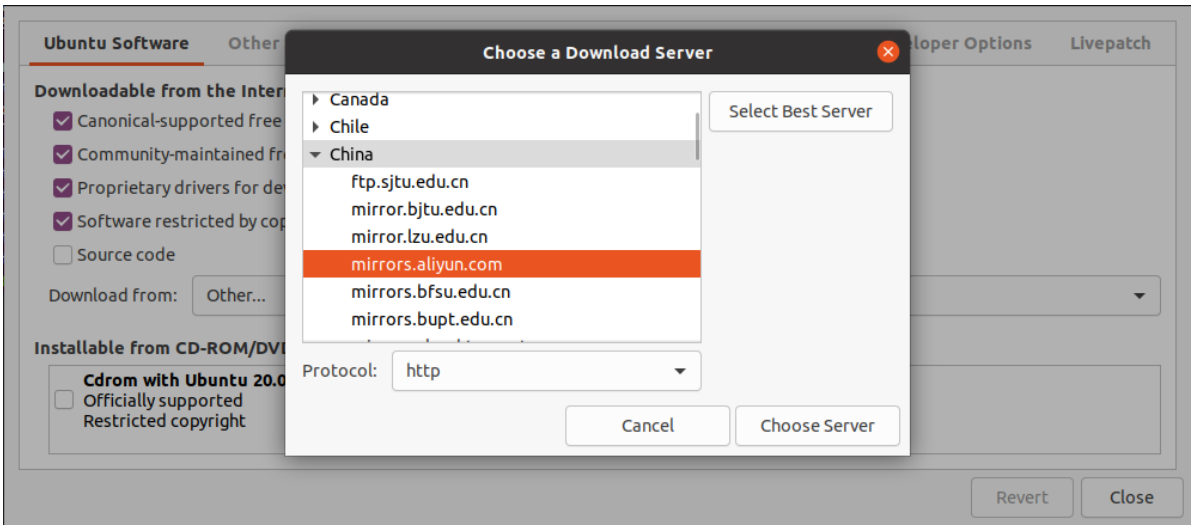
ubuntu配置指南

从系统工具安装到常用软件安装和配置。

系统常用工具

从vim开始配置常用的Ubuntu相关软件。

更换源



搜狗输入法

减少不必要的问题，先清楚ibus，fcitx等框架

```
sudo apt purge ibus*
sudo apt purge fcitx*
sudo apt autoremove
sudo apt-get install fcitx
# download deb文件
sudo dpkg -i sogoupinyin_版本号_amd64.deb
# 如果提示缺少相关依赖
sudo apt -f install
```

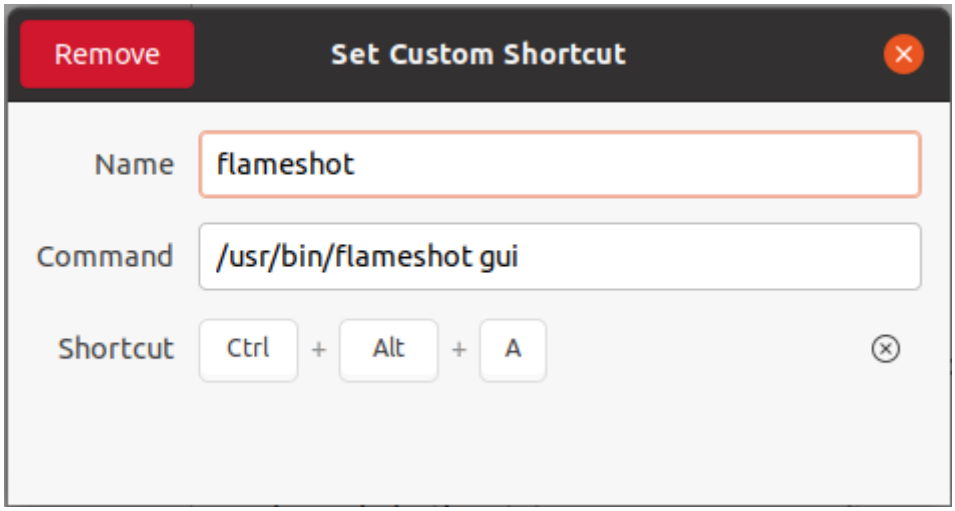
安装和配置部分可参考官方文档 官方安装手册

截图编辑

推荐flameshot截图工具。

```
sudo apt-get install flameshot
```

添加快捷键设置



Git安装和配置

git的安装和全局配置

git安装

```
sudo apt-get install git
```

git配置

```
git config --global user.name "xxx"
git config --global user.email "xxx"
ssh-keygen -C 'email_address' -t rsa
enter
input your password
confirm

#upload
cd ~/.ssh
gedit id_rsa.pub
copy
# 在github ssh界面创建粘贴即可
link:https://github.com/settings/keys
# 测试
ssh -T git@github.com
```

Vim相关

包括vim的安装，vundle安装和配置，vimrc的个人配置文档

vim安装

```
sudo apt-get install vim
```

vundle安装

```
git clone https://github.com/VundleVim/Vundle.vim.git ~/.vim/bundle/Vundle.vim
```

插件的安装需要写在vimrc中，因此需要编写vimrc文件。

vimrc和插件安装

Vim 的全局配置一般在/etc/vim/vimrc或者/etc/vimrc，对所有用户生效。用户个人的配置在~/.vimrc，这里没有太大区别。

我的vimrc配置文件如下(百度网盘链接: https://pan.baidu.com/s/10eMJbke9y5Rf3AdQ13MsnA 密码: s1r2)

```
syntax on
" tab宽度和缩进设置为4
set nu
set tabstop=4
set softtabstop=4
set shiftwidth=4
set mouse=a

set nocompatible

" 设置运行时路径
set rtp+=~/.vim/bundle/Vundle.vim

call vundle#begin()

" 安装的插件
Plugin 'gmarik/Vundle.vim'
Plugin 'kien/rainbow_parentheses.vim'
Plugin 'majutsushi/tagbar'
Plugin 'vim-scripts/ctags.vim'
Plugin 'scrooloose/syntastic'
Plugin 'klen/ctrlp.vim'
Plugin 'scrooloose/nerdtree'
Plugin 'Lokaltog/vim-powerline'
Plugin 'altercation/vim-colors-solarized'
Plugin 'davidhalter/jedi-vim'
Plugin 'klen/python-mode'
Plugin 'nathanaelkane/vim-indent-guides'

Plugin 'MarcWeber/vim-addon-mw-utils'
```

```
Plugin 'tomtom/tlib_vim'
Plugin 'garbas/vim-snipmate'
Plugin 'honza/vim-snippets'
call vundle#end()

" filetype off
filetype plugin indent on
let g:pydiction_location = '~/.vim/bundle/pydiction/complete-dict'
let g:pydiction_menu_height = 15

map <F3> :NERDTreeMirror<CR>
map <F3> :NERDTreeToggle<CR>

let g:snipMate = { 'snippet_version' : 1 }
```

编辑好vimrc文件之后

```
vim
# 安装插件
:PluginInstall
# 部分插件可能已经无法使用，在vimrc注释掉即可
```

Nvidia相关

显卡驱动

```
ubuntu-drivers devices
sudo ubuntu-drivers autoinstall
sudo reboot
nvidia-smi
```

cuda安装

打算安装11.4.0版本,不同版本的安装方式官方均有。CUDA Toolkit所有版本

```
wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin
sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
wget
https://developer.download.nvidia.com/compute/cuda/11.4.1/local_installers/cuda-repo-ubuntu2004-11-4-local_11.4.1-470.57.02-1_amd64.deb
sudo dpkg -i cuda-repo-ubuntu2004-11-4-local_11.4.1-470.57.02-1_amd64.deb
sudo apt-key add /var/cuda-repo-ubuntu2004-11-4-local/7fa2af80.pub
sudo apt-get update
sudo apt-get -y install cuda
```

已安装的版本和显卡驱动不匹配的话，需要卸载已有的cuda

```
sudo apt-get remove cuda
sudo apt autoremove
sudo apt-get remove cuda*
```

设置环境变量

```
sudo vim ~/.bashrc
# type in
export PATH=/usr/local/cuda-11.4/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-11.4/lib64:$LD_LIBRARY_PATH
source ~/.bashrc
nvcc -V
```

cudnn安装

进入官网下载需要的cudnn版本，选择`cuDNN Library for Linux`。

```
tar -xzvf cudnn-11.4-linux-x64-v8.2.4.15.tgz
sudo cp cuda/include/cudnn* /usr/local/cuda-11.4/include
sudo cp cuda/lib64/libcudnn* /usr/local/cuda-11.4/lib64
sudo chmod a+r /usr/local/cuda/include/cudnn*  /usr/local/cuda/lib64/libcudnn*
# confirm
cat /usr/local/cuda/include/cudnn_version.h  | grep CUDNN_MAJOR -A 2
```

OpenCV

包含了opencv_contrib等诸多模块。

下载需要的源码文件。

首先安装cmake等必要的库

```
sudo apt-get install cmake
sudo apt-get install build-essential pkg-config libgtk2.0-dev libavcodec-dev
libavformat-dev libjpeg-dev libswscale-dev libtiff5-dev
```

推荐安装3.4.1，头铁准备安装4.5.3,安装过程基本一致，版本不一致可能导致许多api版本无法使用，需要对应的修改源代码。

boostdesc等文件可能会在后面失败，下载后拷贝到xfeatures2d/src目录下。

链接: https://pan.baidu.com/s/14T0dDs6BRIvU18x0o_ZY7A 密码: 4ptg

```
mkdir build
cd build
sudo cmake -D CMAKE_BUILD_TYPE=Release \
      -D CMAKE_INSTALL_PREFIX=/usr/local \
      -D OPENCV_EXTRA_MODULES_PATH=./../opencv_contrib-4.5.3/modules/ \
      -D WITH_GTK=ON \
      -D OPENCV_GENERATE_PKGCONFIG=YES \
      ..
make -j8
sudo make install

sudo vim /etc/ld.so.conf.d/opencv.conf
# type in
/usr/local/lib
sudo ldconfig
# warnings
sudo ln -sf /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_adv_train.so.8.2.4 /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_adv_train.so.8
sudo ln -sf /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_ops_infer.so.8.2.4 /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_ops_infer.so.8
sudo ln -sf /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_cnn_train.so.8.2.4 /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_cnn_train.so.8
sudo ln -sf /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_adv_infer.so.8.2.4 /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_adv_infer.so.8
sudo ln -sf /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_ops_train.so.8.2.4 /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_ops_train.so.8
sudo ln -sf /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_cnn_infer.so.8.2.4 /usr/local/cuda-11.4/targets/x86_64-
linux/lib/libcudnn_cnn_infer.so.8
sudo ln -sf /usr/local/cuda-11.4/targets/x86_64-linux/lib/libcudnn.so.8.2.4
/usr/local/cuda-11.4/targets/x86_64-linux/lib/libcudnn.so.8
sudo ldconfig
```

```
sudo vim /etc/bash.bashrc
# type in
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig
export PKG_CONFIG_PATH

source /etc/bash.bashrc

pkg-config --modversion opencv4
pkg-config --cflags opencv4
pkg-config --libs opencv4

sudo mv /usr/local/include/opencv4 /usr/local/include/opencv
sudo ln -s /usr/local/include/opencv/opencv2 /usr/local/include/opencv2
sudo ln -s /usr/local/include/opencv/opencv2 /usr/local/include/opencv4
```

Locate

```
sudo apt-get install mlocate
sudo updatedb
//locate eigen3
```

Eigen

SLam等需要的一个库

```
sudo apt-get install libeigen3-dev
cd /usr/local/include
sudo ln -sf ../../include/eigen3/Eigen Eigen
sudo ln -sf ../../include/eigen3/unsupported unsupported
```

Fim

```
sudo apt-get install fim
```

Tmux

```
sudo apt-get install -y tmux
```

个性化设置

vim ~/.tmux.conf后输入配置即可，GitHub上有很多类似的文件，也可以根据需要设置。

我的配置文件如下：

链接: https://pan.baidu.com/s/1gfWdRissQh3t1H0dQTZQeA 密码: sqjk

```
unbind ^b
set -g prefix C-x
bind C-x send-prefix

unbind '"'
bind v splitw -v
unbind %
bind h splitw -h
bind r source-file ~/.tmux.conf \;

unbind C-[
unbind C-]
bind C-n new-session

bind Tab last-window

#set status-interval 1
set-option -g status on
set-option -g status-interval 1
set-option -g status-justify "left"
set-option -g status-left-length 60
set-option -g status-right-length 90
#选择分割的窗格
#bind k selectp -U
#bind j selectp -D
#bind h selectp -L
#bind l selectp -R
#重新调整窗格的大小
#bind ^k resizep -U 10
#bind ^j resizep -D 10
#bind ^h resizep -L 10
#bind ^l resizep -R 10
#swap to panels
bind ^u swapp -U
bind ^d swapp -D

set -g status-fg colour055
set -g status-bg colour032
set -g default-terminal "screen-256color"

# 对齐方式
set-option -g status-justify left

# 左下角
# set-option -g status-left '#[bg=black,fg=green][#[fg=blue]#S#[fg=green]]'
set-option -g status-left-length 20
set-option -g allow-rename off  #do not change your window title automaticly.
set-window-option -g window-status-format '#[fg=colour226,bold]#I:#W_#
{window_panes}'
```

```
## just for mac settings.
## install reattach-to-user-namespace first.
## Copy-paste integration
## Use vim keybindings in copy mode
## Setup 'v' to begin selection as in Vim
# set-option -g default-command "reattach-to-user-namespace -l zsh"
## if your are in osx env , uncomment line up.

# setw -g mode-keys vi
# bind-key -t vi-copy v begin-selection
# bind-key -t vi-copy y copy-pipe "reattach-to-user-namespace pbcopy"
# unbind -t vi-copy Enter
# bind-key -t vi-copy Enter copy-pipe "reattach-to-user-namespace pbcopy"
# bind ] run "reattach-to-user-namespace pbpaste | tmux load-buffer - && tmux
paste-buffer"

if "test ! -d ~/.tmux/plugins/tpm" \
  "run 'git clone https://github.com/tmux-plugins/tpm ~/.tmux/plugins/tpm'"

set-option -g mouse on

# List of plugins
set -g @plugin 'tmux-plugins/tmux-sensible'
set -g @plugin 'nhdaly/tmux-better-mouse-mode'
set -g @plugin 'NHDaly/tmux-scroll-copy-mode'
set -g @plugin 'tmux-plugins/tmux-resurrect'
set -g @plugin 'tmux-plugins/tmux-yank'

set -g status-interval 60
set -g status-justify centre

# setting for tmux-resurrect
set -g @resurrect-strategy-vim 'session'
set -g @resurrect-strategy-nvim 'session'
set -g @scroll-down-exit-copy-mode "on"
set -g @scroll-in-moused-over-pane "on"
set -g @scroll-without-changing-pane "on"
set -g @emulate-scroll-for-no-mouse-alternate-buffer "on"
```

然后激活配置文件

```
tmux source-file ~/.tmux.conf
```

Indicator-multiload

```
sudo apt-get install indicator-multiboard
```

Gnome-tweak-tool

```
sudo apt-get install gnome-tweak-tool
```

Curl

```
sudo apt install curl
# 查看天气
curl wttr.in
curl wttr.in/shanghai
```

类似查看天气的好玩插件很多，不做赘述

Python输出latex符号

例如在matplotlib中需要输入latex语句

```
pip install latex
sudo apt-get install dvipng
sudo apt-get install -y texlive texlive-latex-extra texlive-latex-recommended
sudo apt install cm-super
```

第三方软件及相关配置

Typora 安装和配置

typora安装

```
# or use
# sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys BA300B7755AFCFAE
wget -qO - https://typora.io/linux/public-key.asc | sudo apt-key add -

# add Typora's repository
sudo add-apt-repository 'deb https://typora.io/linux ./'
sudo apt-get update

# install typora
sudo apt-get install typora
```

typora配置和主题安装

Perferences根据需要进行配置即可。

主题从官网获取即可。

百度网盘

网盘地址从官网下载百度网盘后，dpkg -i安装

添加自定义的别名:

```
sudo vim ~/.bashrc
# type in
alias pan="/opt/baidunetdisk/baidunetdisk"
source ~/.bashrc
```

以后可以用在命令行输入pan启动百度网盘。

Anaconda安装配置

Anaconda安装

download 安装包 官网

```
bash Anaconda3-2021.05-Linux-x86_64.sh
# 跟随指示即可，建议选择conda init
```

Anaconda取消自动激活

```
conda config --set auto_activate_base false
```

Anaconda 常用指令

创建环境

```
conda create -n python36 python=3.6
```

导出依赖文件

```
conda env export > requirements.yml
```

`conda activate (xxx)`，`conda deactivate`，`conda remove -n (xxx) --all`指令不做赘述。

VScode

VScode安装和个人配置

VScode 安装

官网下载安装即可。

```
sudo dpkg -i code_1.60.0-1630494279_amd64.deb
```

不需要添加自定义的别名，终端输入`code`即可启动VSCode

VSCode插件

可以安装以下插件。

```
adpyke.codesnap-1.3.4
christian-kohler.path-intellisense-2.4.0
clemenspeters.format-json-1.0.2
coenraads.bracket-pair-colorizer-1.0.61
dbaeumer.vscode-eslint-2.1.25
donjayamanne.githistory-0.6.18
eamodio.gitlens-11.6.0
esbenp.prettier-vscode-8.1.0
grapecity.gc-excelviewer-3.0.44
gruntfuggly.todo-tree-0.0.214
hookyqr.beautify-1.5.0
kisstkondoros.vscode-gutter-preview-0.27.1
mechatroner.rainbow-csv-1.9.1
mhutchie.git-graph-1.30.0
ms-python.python-2021.9.1230869389
ms-python.vscode-pylance-2021.9.1
ms-toolsai.jupyter-2021.8.2041215044
ms-toolsai.jupyter-keymap-1.0.0
ms-vscode.cpptools-1.6.0
oderwat.indent-rainbow-8.0.0
seanwu.vscode-qt-for-python-1.1.4
usernamehw.errorlens-3.4.0
visualstudioexptteam.vscodeintellicode-1.2.14
vscode-icons-team.vscode-icons-11.6.0
xyz.local-history-1.8.1
zhucy.project-tree-0.3.0
```

查看已经安装的插件集

```
code --list-extensions
```

CLion(C++)

ubuntu16以后可以从命令行一键安装

```
sudo snap install clion --classic
```

仙人指路

CLion_MJ.jar安装包

链接: https://pan.baidu.com/s/1QCXCikzHd09uEmf-77VYcg 密码: fi4f

常用的指令

查看序列号

```
lsblk --nodeps -no serial /dev/sda
# W9AFY5ZD
```

查看内存使用

```
top
```