

# Deepin Linux 使用笔记

JackLovel

2020 年 3 月 31 日

# 目录

第一章 安装	1
1.1 下载镜像 . . . . .	1
1.2 制作启动盘镜像 . . . . .	1
1.3 分区建议 . . . . .	2
第二章 系统配置	3
2.1 更新源 . . . . .	3
第三章 基本软件的配置	4
3.1 zsh . . . . .	4
3.2 emacs . . . . .	5
3.3 jdk . . . . .	6
3.4 python . . . . .	7
3.5 qt creator . . . . .	9
3.6 latex . . . . .	10

# 第一章 安装

## 1.1 下载镜像

可以到[清华源](#)下载最新的系统镜像

<https://mirrors.tuna.tsinghua.edu.cn/deepin-cd/>

## 1.2 制作启动盘镜像

1. 如果在 window 系统下, 可以使用 Rufus, 选好镜像后, 分区类型选 GPT, 刻录模式为 DD

Rufus 下载地址:

<https://share.weiyun.com/51nzNxs>

2. 如果在 linux 系统下, 可以使用 dd 命令。

```
$ sudo fdisk -l

# 我的U盘分区: /dev/sdc
$ umount /dev/sdc1
$ sudo mkfs.vfat /dev/sdc -I

# 我本地 deepin 镜像的完整路径: /home/gog/下载/deepin-15.11-amd64.iso
# U盘分区: /dev/sdc
$ sudo dd bs=4M if=/home/gog/下载/deepin-15.11-amd64.iso of=/dev/sdc status=progress
```

### 1.3 分区建议

这是一个建议的分区，而不是必须的，仅供参考，这里以 250G 为例：

挂载点	分区大小 (G)	占比 (百分比)
/	100	40
/home	100	40
交换分区 (/swap)	10	
/boot/efi	0.2	

这里分区有几点要说明一下：

- 根分区和 home 分区占大比重，其中 home 你可以在最后分，把剩下的所有的都给它。
- 这里的换算：1GB = 1000MB

## 第二章 系统配置

### 2.1 更新源

- 修改更新源

```
// 以阿里源为例

$ sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
$ sudo deepin-editor /etc/apt/sources.list

* 将其中的 http://packages.deepin.com/deepin 替换成 http://mirrors.aliyun.com/deepin
```

- 更新

```
# 更新软件列表
$ sudo apt update

# 升级软件
$ sudo apt upgrade
```

## 第三章 基本软件的配置

### 3.1 zsh

由于系统原来 bash shell 不怎么好用，所以这里我们推荐使用 zsh

- 安装 zsh:

```
$ sudo apt-get install -y zsh
$ sh -c \
"$(curl -fsSL https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"

# 将 zsh 更改为默认的 shell
$ chsh -s /bin/zsh
```

- 安装 percol

```
# 如果 pip 没有安装的话
$ sudo apt install -y python-pip

$ sudo pip install percol
$ deepin-editor ~/.zshrc
```

- 添加配置

```
function exists { which $1 &> /dev/null }

if exists percol; then
    function percol_select_history() {
        local tac
        exists gtac && tac="gtac" || { exists tac && tac="tac" || { tac="tail -r" } }
        BUFFER=$(fc -l -n 1 | eval $tac | percol --query "$LBUFFER")
        CURSOR=$#BUFFER      # move cursor
        zle -R -c            # refresh
    }

    zle -N percol_select_history
    bindkey '^R' percol_select_history
fi
```

## 3.2 emacs

下面介绍二种方式安装 emacs:

1. 每一种方法, 比较简单, 但是 emacs 版本还是比较低的。

```
$ sudo apt install -y emacs
```

```
// 查看版本
```

```
$ emacs --version
```

2. 第二种方法, 相对于第一种方法来说, 略微复杂, 比如: 依赖问题; 但是软件版本还是比较高的。

- 下载源码

```
https://www.gnu.org/software/emacs/download.html#gnu-linux
```

- 解压

```
$ tar xvf emacs-26.2.tar.gz // 解压, 并切换到解压后的目录
```

- 安装依赖

```
$ sudo apt-get install build-essential \  
texinfo libx11-dev libxpm-dev libjpeg-dev \  
libpng-dev libgif-dev libtiff-dev libgtk2.0-dev \  
libncurses-dev libxpm-dev automake autoconf
```

- 编译及安装

```
$ ./configure --with-mailutils  
$ sudo make && sudo make install
```

- 检测

```
$ emacs --version
```

### 3.3 jdk

这里安装的是 oracle jdk, 所以到 oracle 官网下载 jdk

- 下载

```
https://www.oracle.com/technetwork/java/javase/downloads/jdk12-downloads-5295953.html
```

- 解压

```
$ tar xvf jdk-12.0.2_linux-x64_bin.tar.gz
```

- 添加配置, 将下面的内容写入 `/.zshrc` 或者 `/.bashrc`

```
export JAVA_HOME= 此处填写jdk的绝对路径
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

- 检测

```
$ source ~/.zshrc

$ java -version

$ javac
```



## 3.4 python

- pip 安装

```
# python2
$ sudo apt install python-pip
$ sudo pip --version

# python3
$ sudo apt install python3-pip
$ sudo pip3 --version
```

- pypi 配置

```
$ mkdir ~/.pip
$ cd ~/.pip
$ touch pip.conf
$ deepin-editor ~/.pip/pip.conf
```

然后添加下面的内容:

```
[global]
index-url = http://pypi.douban.com/simple
[install]
trusted-host=pypi.douban.com
```

- ipython

```
# python2
$ sudo apt install -y ipython

# python3
$ sudo apt install -y ipython3
```

- pyenv

python 版本管理工具

```
# 下载 pyenv
$ git clone https://github.com/pyenv/pyenv.git ~/.pyenv

# 配置环境
$ echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.zshrc
$ echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.zshrc
$ echo 'eval "$(pyenv init -)"' >> ~/.zshrc

# 使配置生效
$ source ~/.zshrc
```

```
# 检测
$ pyenv --help

# deepin 系统中不建议 删除 原有的python版本，具体原因这里就不细说了。
# 这里我举一个 安装 python3.7.4 版本的过程：
$ pyenv install -v 3.7.4
$ pyenv global 3.7.4 # 设置系统中 python 版本
$ pyenv versions # 查看当前系统 python 的版本
```

### 3.5 qt creator

- 下载

[qt-opensource-linux-x64-5.8.0.run](#) 下载地址

- 安装

```
$ chmod +x qt-opensource-linux-x64-5.8.0.run
$ ./qt-opensource-linux-x64-5.8.0.run
```

- 解决中文输入

```
$ cd /usr/lib/x86_64-linux-gnu/qt5/plugins/platforminputcontexts

# 我的 qt安装目录: ~/Qt5.8.0
$ cp libfcitxplatforminputcontextplugin.so \
  ~/Qt5.8.0/5.8/gcc_64/plugins/platforminputcontexts

$ cp libfcitxplatforminputcontextplugin.so \
  ~/Qt5.8.0/Tools/QtCreator/lib/Qt/plugins/platforminputcontexts
```

## 3.6 latex

tex live 安装

- 下载

```
// 下载 texlive2019.iso  
https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/Images/
```

- 安装 latex:

```
// 首先, 解压 镜像  
// 然后安装  
$ chmod +x install-tl  
$ sudo ./install-tl
```

添加下面的内容

```
export PATH=/usr/local/texlive/2019/bin/x86_64-linux:$PATH  
export MANPATH=/usr/local/texlive/2019/texmf-dist/doc/man:$MANPATH  
export INFOPATH=/usr/local/texlive/2019/texmf-dist/doc/info:$INFOPATH
```

- 测试

```
$ source ~/.zshrc  
$ tex -v
```