Ubuntu 使用笔记

JackLovel

2020 年 9 月 7 日

目录

第一章　序

当你看本书时，你可能学起来有困难，这里我推荐两本不错的书《Linux 系统与网络管理》(姜大庆、周建、邓荣 编) 和 《Linux 命令行大全》（William E.Shotts，Jr. 编，郭光伟，郝记生译)。你可以从这两本书选一本书看完，作为自己的前置的知识。然后再来看这本书。

这本书的结构?
第一章，系统安装，可以仔细看；
第二章，系统源的配置；
第三章，软件安装，这部分可以挑着看，需要的时候搜索即可；
第四章，一些好玩的部分。

本书何时更新完？或许没有一个最终版本，我会不时的更新和纠正里面的内容。

第二章　安装

2.1　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/

2.2　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 ubuntu 镜像的完整路径：/home/gog/下载/ubuntu-20.04.1-desktop-amd64.iso
   # U盘分区： /dev/sdc
   $ cd /home/gog/下载/
   $ sudo dd bs=4M if=ubuntu-20.04.1-desktop-amd64.iso of=/dev/sdc status=progress
   ```

2.3 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）	8	
efi 分区（/boot/efi）	0.2	
/boot	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 [1]

[1]在数学换算里 1G = 1024M，但是这里是 1000M

第三章　系统配置

3.1　更新源

- 修改更新源 [1]

```
// 以阿里源为例
$ sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
$ sudo nano /etc/apt/sources.list

* 仅需要将其中的 http://cn.archive.ubuntu.com/ubuntu/ 替换成
  http://mirrors.aliyun.com/ubuntu/ 即可
```

- 更新

```
# 更新软件列表
$ sudo apt update

# 升级软件
$ sudo apt upgrade
```

- 补充：清华源添加方法

[1]类似的像 ubuntu 的发行版本，也是通过替换相应的源的地址就可以了。

第四章　基本软件的配置

4.1　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.github.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
```

4.2 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
```

4.3 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
```

4.4 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
```

4.5 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
```

4.6 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
```

4.7 mysql

- 安装:

```
$ sudo apt install mysql-server -y
```

- 初始化数据库

```
$ sudo mysql_secure_installation

// 参考
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: N
Please set the password for root here.

New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : N

 ... skipping.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
```

```
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : N

 ... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y
Success.

All done!
```

- 创建数据库

```
$ sudo mysql -hlocalhost -uroot -p

// 更改加密方式
$ ALTER USER 'root'@'localhost' IDENTIFIED BY 数据库root字符串 PASSWORD EXPIRE NEVER;

// 更改密码
$ ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 数据库root字符串;

// 刷新权限
$ FLUSH PRIVILEGES;
```

- 验证

```
$ mysql -hlocalhost -uroot -p
```

4.8 docker

- 安装:

```
// 添加Docker官方GPG key
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

// 添加仓库
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable"

// 更新
$ sudo apt-get update
```

```
// 安装 docker-ce
$ sudo apt-get install docker-ce

// 检查是否安装成功
$ sudo docker run hello-world

// 设置用户权限，把当前用户添加 docker 用户组里
$ sudo usermod -a -G docker $USER
```

- 添加国内源登录 阿里源 docker 镜像服务，然后点击镜像中心-> 镜像加速器

```
// 仅供参考，我的阿里源加速器配置，有可能加速器地址不一致
$ sudo mkdir -p /etc/docker
$ sudo tee /etc/docker/daemon.json <<-'EOF'
  {
    "registry-mirrors": ["https://00o606tc.mirror.aliyuncs.com"]
  }
  EOF
$ sudo systemctl daemon-reload
$ sudo systemctl restart docker
```

- 测试来下载一个镜像，来测试一下，添加的国内源是否生效

```
// 拉取 ubuntu 镜像
$ docker pull ubuntu

// 运行以 ubuntu 为镜像，并命名为 ubuntu-test 的容器
$ docker run -itd --name ubuntu-test ubuntu
```

第五章　其它的软件

5.1　Rust 语言

- 安装[1]：

```
$ curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

- 配置 Cargo 环境：
  Cargo 用于包管理和编译程序的工具，将下面的内容添加到 ~/.bashrc 文件中：

```
export PATH=$HOME/.cargo/bin:$PATH
```

- 使生效

```
$ source ~/.bashrc
```

- 检测

```
# 检测 cargo 版本
$ cargo --version

# 检测 rust 版本
$ rustc --version
```

[1]官网安装教程：https://www.rust-lang.org/learn/get-started

5.2 Go 语言

- 安装[2]：

```
$ sudo add-apt-repository ppa:longsleep/golang-backports
$ sudo apt update
$ sudo apt install golang-go
```

- 添加国内源

```
export GOPROXY=https://goproxy.cn
```

- 使生效

```
$ source ~/.bashrc
```

- 检测

```
$ go version
```

- 例子 hello.go:

```
package main
import "fmt"
func main() {
	fmt.Printf("你好世界\n")
}

// 在命令行里运行：go run hello.go
```

[2]官网安装教程：https://github.com/golang/go/wiki/Ubuntu

第六章　命令补充

6.1　打印系统相关的信息

- 安装：

```
# ubuntu
$ sudo apt-get install -y screenfetch
```

- 运行：

```
$ sudo screenfetch
```

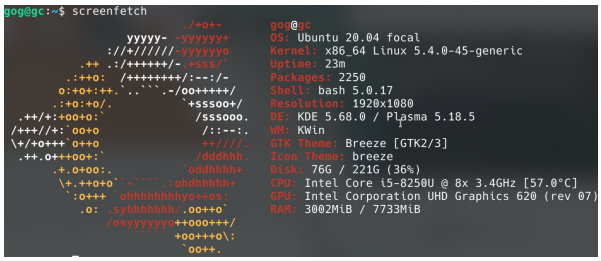

图 6.1：运行结果

6.2 apt

debian 系[1]用的最多的命令，是必需的命令之一。

- 常用命令

```
# 更新软件列表
$ sudo apt-get update

# 升级软件
$ sudo apt-get updrade

# 删除包
$ sudo apt-get remove <package_name>

# 自动卸载软件，及其有依赖相关的软件包
$ sudo apt-get autoremove

# 安装包
$ sudo apt-get install -y <package_name>

# 支持模糊搜索
$ sudo apt-get search <package_name>

# 解决依赖
$ sudo apt install -f
```

- ppa 源[2]

[1]常见的 debian 系：ubuntu，deepin，kubuntu，kail，xubuntu

[2]官网：https://launchpad.net/

6.3 snap

你可以通过snap 应用商店来查找软件，也可以通过命令行来查找。

- 常用命令

  snap 是一个包管理器，类似于 apt 可用于软件的安装、查找、卸载

```
$ snap help
$ snap search <包名> // 模糊查找
$ snap install <包名>
$ snap remove <包名>
$ snap list
```

- 例子：安装 vscode

```
$ snap install code
```

第七章　游戏

7.1　GOG

- 安装 lgogdownloader：

  由于 GOG 平台没有 Linux 客户端，所以只能通过命令行来下载。

```
$ sudo apt install -y lgogdownloader

# 登陆 Gog 账号
$ sudo lgogdownloader --login

# 查看自已的游戏库
$ sudo lgogdownloader --list

# 下载游戏，下载完成后，运行脚本即可。
$ sudo lgogdownloader --download --platform 4 --game <游戏名称>
```

7.2 steam

与 gog 平台相比 steam 有自己的 Linux 客户端，并且速度游戏下载速度更快。

- 下载客户端

  下载 steam Linux 客户端

图 7.1: 下载页面

- 安装

```
$ sudo dpkg -i steam_latest.deb
```

参考文献