

Linux系统配置指南

Linux的配置指南

包括的Linux系统有Arch系（Arch、Manjaro）和Debian系（Ubuntu、deepin）

本仅提供常见Linux OS的配置方法，不提供系统的安装方法

如发现错误，欢迎指正

目录

- [Linux系统配置指南](#)
 - [目录](#)
 - [基本配置](#)
 - [软件源](#)
 - [pacman](#)
 - [yay](#)
 - [时间](#)
 - [输入法](#)
 - [fcitx](#)
 - [ibus](#)
 - [环境搭建](#)
 - [Python](#)
 - [pip](#)
 - [Anaconda](#)
 - [Go](#)
 - [Latex](#)
 - [工具及终端配置](#)
 - [vim/neovim配置](#)
 - [zsh配置](#)
 - [安装zsh](#)
 - [安装oh-my-zsh](#)
 - [ranger配置](#)
 - [st & dwm](#)
 - [美化](#)
 - [Gnome美化](#)
 - [KDE Plasma美化](#)
 - [Dock](#)
 - [主题](#)
 - [Plasma样式](#)
 - [图标](#)
 - [光标](#)
 - [欢迎屏幕](#)
 - [Linux必安装的依赖以及应用](#)

- [常见问题](#)
 - [Alacritty终端模拟器](#)
 - [在Gnome中使用fxitx](#)
 - [manjaro系统编译LaTeX生成的PDF无法显示中文](#)

基本配置

软件源

pacman

切换到中国源(Manjaro)

```
sudo pacman-mirrors -i -c China -m rank
```

选择速度最快的一个软件源

然后在 `/etc/pacman.conf` 文件末尾添加以下内容：

```
[archlinuxcn]
SigLevel = Never
Server = https://mirrors.tuna.tsinghua.edu.cn/archlinuxcn/$arch
```

运行以下命令可以更新软件源并更新软件

```
sudo pacman -Syyu
```

yay

```
yay --aururl "https://aur.tsinghua.edu.cn" --save
```

修改的配置文件位于 `~/.config/yay/config.json`，还可通过以下命令查看修改过的配置：

```
yay -P -g
```

时间

在windows和linux双系统的情况下会出现linux系统时间比当地时间快8小时的情况

解决：

```
sudo timedatectl set-local-rtc true
```

输入法

fcitx

安装fcitx框架及输入法：

```
sudo pacman -S fcitx-im
sudo pacman -S fcitx-configtool
sudo pacman -S fcitx-googlepinyin
sudo pacman -S fcitx-rime
sudo pacman -S rime-double-pinyin
sudo pacman -S kcm-fcitx # KDE Config Module for Fcitx
```

然后将 `default.custom.yaml` 文件复制到 `~/.config/fcitx/rime/` 目录下。

在家目录下创建文件 `.xprofile`，写入以下内容：

```
export LANG=zh_CN.UTF-8
export LC_ALL=zh_CN.UTF-8
export GTK_IM_MODULE=fcitx
export QT_IM_MODULE=fcitx
export XMODIFIERS="@im=fcitx"
```

ibus

```
sudo pacman -S ibus-fime
sudo pacman -S ibus-pinyin
```

在家目录下创建文件 `.xprofile`，写入以下内容：

```
export LANG=zh_CN.UTF-8
export LC_ALL=zh_CN.UTF-8
export GTK_IM_MODULE=ibus
export QT_IM_MODULE=ibus
export XMODIFIERS="@im=ibus"
```

环境搭建

Python

pip

可以安装 Python 软件包的 PyPA 工具。

```
# Ubuntu
sudo apt install python3-pip

# Arch
sudo pacman -S python-pip

#查看pip版本
pip -V
```

添加pip源：复制 `.pip/` 文件夹到家目录。

Anaconda

[Anaconda官网](#)

[Anaconda下载地址](#)

[Miniconda下载地址](#)

[Anaconda清华源](#)

或者运行：

```
# 以下安装方式不推荐，配置起来会很麻烦
sudo pacman -S anaconda
```

添加conda源，将以下内容复制到 `.condarc` 文件下。

```
channels:
  - defaults
show_channel_urls: true
channel_alias: https://mirrors.tuna.tsinghua.edu.cn/anaconda
default_channels:
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/pro
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
custom_channels:
  conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

conda的相关应用

```
# lazygit
conda install -c conda-forge lazygit
```

Go

安装Golang的命令：

```
# 安装Golang
sudo pacman -S go
# 设置`GOPROXY`
go env -w GOPROXY=https://goproxy.cn,direct
```

设置Go语言环境，需要添加 `GOPATH` 和 `GOROOT` 到环境变量。

```
export GOPATH=$HOME/Go/bin
export GOROOT=/usr/lib/go # Golang的安装目录
```

Latex

TeX Live 是一个完整、功能强大的 TeX 发布版本，包含了主要的 TeX 相关程序、宏和字体，官方软件仓库收录了它。老的(停止开发) teTeX 发布版本位于 AUR

[TexLive ArchWiki](#)

```
sudo pacman -S texlive-most
```

工具及终端配置

vim/neovim配置

vim: 只需将配置文件 `.vim` 放到家目录即可（配置文件名为 `.vimrc`）

neovim: 须将其配置文件放到 `~/.config/nvim/` 目录下（配置文件名为 `init.vim`）

vim-plug安装

在安装vim-plug之前确保python以及相关库已经安装

[Jump to Python的环境搭建](#)

```
# vim
curl -fLo ~/.vim/autoload/plug.vim --create-dirs \
  https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim

# neovim
curl -fLo ~/.local/share/nvim/site/autoload/plug.vim --create-dirs \
  https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim

# 如果拒绝连接, 编辑/etc/hosts文件
sudo echo 199.232.28.133 raw.githubusercontent.com >> /etc/hosts

# 安装插件的依赖库
pip install pynvim --upgrade
sudo pacman -S nodejs
sudo pacman -S npm

# 最后打开neovim执行命令
:PlugInstall

# 如果打开.py文件报错, 执行
:UpdateRemotePlugins
```

zsh配置

安装zsh

```
# Arch
sudo pacman -S zsh

# Ubuntu
sudo apt install zsh
```

修改默认Shell为zsh

```
chsh -s /usr/bin/zsh
```

安装oh-my-zsh

```
curl -Lo install.sh
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh && sh
install.sh

# Gitee
sh -c "$(curl -fsSL https://gitee.com/shmhlly/oh-my-zsh-
install.sh/raw/master/install.sh)"

# or
sh -c "$(wget -O- https://gitee.com/shmhlly/oh-my-zsh-
install.sh/raw/master/install.sh)"
```

[oh-my-zsh皮肤](#)

```
# typewritten
git clone https://github.com/reobin/typewritten.git
$ZSH_CUSTOM/themes/typewritten

ln -s "$ZSH_CUSTOM/themes/typewritten/typewritten.zsh-theme"
"$ZSH_CUSTOM/themes/typewritten.zsh-theme"

# powerlevel10k
git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-
~/.oh-my-zsh/custom}/themes/powerlevel10k

# 配置powerlevel10k命令
p10k configure
```

oh-my-zsh插件

```
# zsh-autosuggestions
git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-
my-zsh/custom}/plugins/zsh-autosuggestions

# zsh-syntax-highlighting
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git ${ZSH_CUSTOM:-
~/.oh-my-zsh/custom}/plugins/zsh-syntax-highlighting

# autojump
git clone git://github.com/wting/autojump.git ~/autojump && ./autojump/install.py
```

ranger配置

[ranger的Github页面](#)

包括的插件和命令有：

- ranger-devicons
- extractor
- extractorhere
- compress

复制ranger目录到 `~/ .config/` 目录下

```
# 生成ranger配置文件命令
ranger --copy-config=all

# 修改默认编辑器命令
select-editor # Ubuntu
```

st & dwm

[suckless官网](#)

分别进入两个目录下，输入以下命令可完成编译及安装：

```
make
sudo make clean install
```

其中，dwm需复制 `dwm.desktop` 文件到 `/usr/share/xsessions` 目录下，并将 `.dwm` 目录复制到家目录中。

DWM依赖

```
sudo pacman -S xorg xorg-server # xsetroot命令
sudo pacman -S feh # 壁纸管理
sudo pacman -S picom # 淡入淡出、半透明、阴影等视觉效果
sudo pacman -S trayler # 系统托盘
sudo pacman -S dmenu
```

美化

Gnome美化

```
# gnome美化面板
sudo pacman -S gnome-tweak-tool

# 用chromium配置gnome的工具
sudo pacman -S chrome-gnome-shell
```

插件推荐

- Dash to Dock
- Dash to Panel
- ArcMenu
- Application Menu
- User Themes
- Pop Shell
- Open Weather

KDE Plasma美化

Dock

```
# kde dock
sudo pacman -S latte-dock

# plank dock
sudo pacman -S plank

# 基于 OpenGL的混合型窗口管理器
sudo pacman -S compiz
```

之后运行 `latte-dock` 或 `plank` 开启。
这两种Dock会默认自启动。

主题

Plasma样式

- breeze2
- breeze2 Dark
- McMojave
- cherry
- Qogir-dark
- ChromeOS

图标

- Tela
- Tela Dark

光标

- Breeze 微风
- Breeze 微风亮色
- DeepinDark Cursors
- DeepinWhite Cursors

欢迎屏幕

- BeautifulTreeAnimation
- Harmaony
- McMojave
- QuarksSplashDark
- QuarksSplashDarkLight

Linux必安装的依赖以及应用

```
# =====基础依赖=====

# 设备驱动
sudo pacman -S net-tools
sudo pacman -S xorg xorg-server
```



```
sudo pacman -S xf86-input-synaptics # 触摸板驱动
sudo pacman -S xf86-input-libinput
sudo pacman -S xf86-video-intel # intel显卡驱动
sudo pacman -S xf86-video-ati # amd显卡驱动
sudo pacman -S alsa-utils pulseaudio pulseaudio-alsa # 声音管理器
```

底层应用

```
sudo pacman -S w3m # 终端浏览器
sudo pacman -S ntfs-3g # 识别windows分区
sudo pacman -S make cmake # 编译工具
```

=====应用程序=====

Arch / Manjaro

```
sudo pacman -S dolphin
sudo pacman -S nmtui # 修改静态IP地址
sudo pacman -S zsh
sudo pacman -S fish
sudo pacman -S git
sudo pacman -S yay # Arch的AUR
```

终端模拟器

```
sudo pacman -S gnome-terminal # Gnome默认终端
sudo pacman -S kconsole # KDE默认终端
sudo pacman -S alacritty
sudo pacman -S yakuake
sudo pacman -S tilda
```

终端应用

```
sudo pacman -S htop
sudo pacman -S ranger
sudo pacman -S lazygit
```

窗口管理器

```
sudo pacman -S i3
sudo pacman -S polybar
```

```
sudo pacman -S code # vscode
sudo pacman -S visual-studio-code-bin # vscode可同步版
sudo pacman -S netease-cloud-music # 网易云音乐
sudo pacman -S kdenlive # 视频剪辑软件
sudo pacman -S gimp # 修图软件
sudo pacman -S simplescreenrecorder # 录屏软件
sudo pacman -S libreoffice-fresh-zh-cn # 中文版libreoffice
```

PDF浏览器

```
sudo pacman -S okular
sudo pacman -S evince
```

Latex编辑器

```
sudo pacman -S texstudio
sudo pacman -S texworks
```

Wechat

```
yay -S deepin-wine-wechat
yay -S deepin.com.wechat2
sudo pacman -S electronic-wechat-git
sudo pacman -S electron-wechat
yay -S wechat-devtools
```

```
# LaTeX suit
sudo pacman -S texlive-core texlive-langchinese texlive-latexextra

# QQ & TIM
yay -S deepin-wine-tim
yay -S deepin-wine-qq
```

常见问题

Alacritty终端模拟器

问题：

当输入clear时出现 'alacritty': unknown terminal type. 时

解决办法：

修改环境变量，在/etc/profile中添加如下：

```
$ export TERMINFO=/usr/share/terminfo
重启系统生效。
```

在Gnome中使用fcitx

问题：

将ibus卸载后安装fcitx后无法打字

解决办法：

在 /etc/environment 文件中输入以下内容：

```
GTK_IM_MODULE=fcitx
QT_IM_MODULE=fcitx
XMODIFIERS="@im=fcitx"
```

manjaro系统编译LaTeX生成的PDF无法显示中文

问题：

manjaro系统编译LaTeX生成的PDF无法显示中文

解决办法：

原因是okular软件使用软件包poppler-data解析pdf中的中文

```
sudo pacman -S poppler-data
```