搭建属于你的地震数据处理Ubuntu平台

一些常见的问题解决方案以及实用提示

学校：成都理工大学
学院：地球物理学院
作者：李子霏

2021.06.20-

说明

该文档为在使用ubuntu系统作为地震数据处理平台过程中遇到的一些问题和常用软件安装，根据该文档可以搭建ubuntu系统安装常用的地震数据处理编程软件Madagascar和seismic unix，以及使用ubuntu终端直接进行python、matlab、c/cpp/cu语言的书写和运行。对于想要使用ubuntu作为主系统进行工作的同志，强烈建议对zshell，neovim进行配置，否则在使用过程中会耽误很多时间。

Chapter1:系统软件篇

安装ubuntu

- 主分区

主分区是Linux硬盘中最基本的分区类型。在硬盘上，您可以创建最多4个主分区。每个主分区都可以安装一个操作系统。如果您想安装多个操作系统，则需要一个主分区来保存每个操作系统。

另外，一个主分区可以变成扩展分区或逻辑分区，但是这将减少可用的主分区数目。扩展分区是一个特殊的主分区，它可以划分多个逻辑分区。扩展分区本身不能保存文件或数据，只能用来创建逻辑分区。

- 逻辑分区

逻辑分区是一种不同于主分区的分区类型，它不存在于硬盘上的分区表中。逻辑分区只能建在一个扩展分区上，而扩展分区则必须是硬盘上4个主分区之一。逻辑分区不需要格式化为文件系统，只需要被分配了一个文件系统类型的ID。逻辑分区可存储文件和数据，就像主分区一样。

- 分区大小设置：

一般来说安装ubuntu，选择手动分区的情况下只需要根据这个来分区就行

分区名称	分区格式	分区类型	大小	说明
/swap	逻辑分区	/swap或交换分区	电脑内存	虚拟内存，当内存小于16G时可以设成内存两倍，大于16G时和内存一样大就行
/home	主分区	ext4	D盘	存放数据以及安装软件可以在这，这个分区可以最后设置然后剩余的所有内存都给home
/	逻辑分区	ext4	C盘	系统盘，对于新手强烈建议这个分区给大一点，因为软件默认都是安装在这， 但是ubuntu指定安装路径并不像windows那么简单， 很多新手不知道如何指定安装路径到home路径下。
/boot	逻辑分区	ext4	0.5-2G	存放一些日志文件和启动引导项，根据你的ubuntu系统盘大小视情况而定

Zshell!

安装zshell

```
sudo apt-get install zsh
```

oh-my-zsh配置

```
git clone git://github.com/robbyrussell/oh-my-zsh.git /home/lzf/softwares/oh-my-zsh
cp /home/lzf/softwares/oh-my-zsh/templates/zshrc.zsh-template ~/.zshrc
```

然后记得在.zshrc文件中将oh-my-zsh文件夹路径定位。默认路径是~/.oh-my-zsh

neovim!

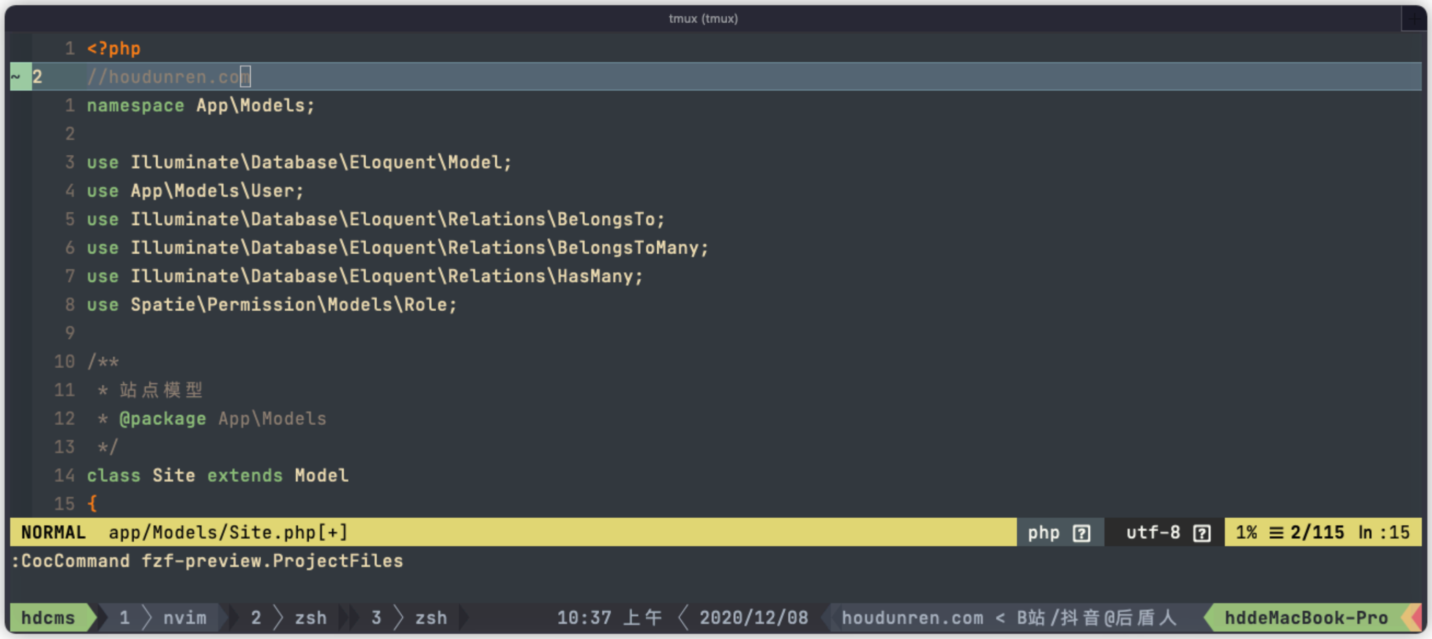
install

```
sudo apt-get install neovim
```

主题插件配置

快速安装

这是 向军大叔 使用的 neovim 配置，本章可以帮助你快速配置好用的 nvovim 环境



```
tmux (tmux)
<?php
//houdunren.com
namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use App\Models\User;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\BelongsToMany;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Spatie\Permission\Models\Role;

/**
 * 站点模型
 * @package App\Models
 */
class Site extends Model
{
NORMAL  app/Models/Site.php[+]                php   utf-8   1% ≡ 2/115 ln :15
:CocCommand fzf-preview.ProjectFiles

hdcms  1 〉nvim  2 〉zsh  3 〉zsh    10:37 上午 〈 2020/12/08  houdunren.com < B站/抖音@后盾人  hddeMacBook-Pro
```

安装软件

环境依赖

下面是依赖的软件环境，请确定系统中已经安装成功

- npm
- yarn
- python
- neovim
- vim-plug
- ranger

首先安装环境需要的软件

```
# mac
brew install node python3 yarn ranger python-pip

# manjaro
sudo pacman -Sy node python3 yarn ranger python-pip

# ubuntu
sudo apt install node python3 yarn ranger python-pip

# centos
sudo yum install node python3 yarn ranger python-pip
```

然后安装 neovim 需要软件包

```
python -m pip install pynvim

pip3 install --user --upgrade neovim

pip3 install ranger-fm
```

vim-plug是一款非常轻量又高效的 vim 插件管理工具。它支持全异步、多线程并行安装插件，支持 git 分支、标签等，可以对插件进行回滚更新、还支持**按需加载**插件(On-demand loading)，可以指定对特定文件类型加载对应 vim 插件，大大加快了 vim 启动时间。

vim-plug

可通过官网查看安装细节 https://github.com/junegunn/vim-plug，因为是国外服务器所以你要多试几次。为了帮助大家正常下载，下面的的链接地址已经放在后盾人 CDN 服务器上了。

```
curl -fLo ~/.local/share/nvim/site/autoload/plug.vim --create-dirs \
    https://houdunren-video.oss-cn-hangzhou.aliyuncs.com/soft/plug.vim
```

执行安装

下面列出的常用系统的安装方法，其他系统参考官方文档进行安装

```
# MAC
brew install neovim

# manjaro
sudo pacman -Sy neovim

# ubuntu
sudo apt install neovim

# CENTOS 8
yum install neovim
```

注销并重新登录后执行 `nvim` 就可以打开软件了

安装配置

配置还是比较简单的，下载包后执行脚本，再次打开 neovm 时将自动安装插件

下载项目

clone 项目

```
git clone https://gitee.com/houdunwang/nvim.git ~/.config/nvim
```

安装插件

打开 nvim 执行以下命令安装 coc 扩展

```
CocInstall coc-css coc-explorer coc-html coc-snippets coc-ember coc-json coc-emmet coc-tsserver coc-highlight coc-prettier coc-vetur coc-git cc

CocCommand eslint.showOutputChannel
```

安装后执行

安装 intelephense

```
npm i intelephense -g
```

进入 `~/.vim/plugged/bracey.vim` 执行以下命令，用于生成 liver-server 环境

```
cd ~/.vim/plugged/bracey.vim
npm install --prefix server
```

按键定义

Leader 键定义为了空格，下面是自定义的按键说明

移动定位

热键	说明	模式
mm	添加注释	
shift+k	上移 5 行	
shift+j	下移 5 行	
gd	转到类、函数定义	
gf	跳转到文件	
ctrl+j	行首	编辑
ctrl+k	行尾	编辑

文件操作

热键	说明
rc	打开当前目录
rp	打开项目目录
fp	项目文件检测
fb	显示 Buffers 文件
fg	显示文件 GIT 状态
C-j	保存
C-k	最近打开的文件
leader+f	fzf 项目文件模糊搜索
leader+b	fzf Buffer 文件搜索

浮动窗口

热键	说明
tl	打开浮动窗口
tr	打开 Ranger 浮动窗口
td	连接 homestead 数据库
ctrl+h	隐藏浮动窗口
ctrl+n	切换上个浮动窗口

其他操作

热键	说明
:Bracey	浏览器同步插件，类似 vscode 中的 liver-server

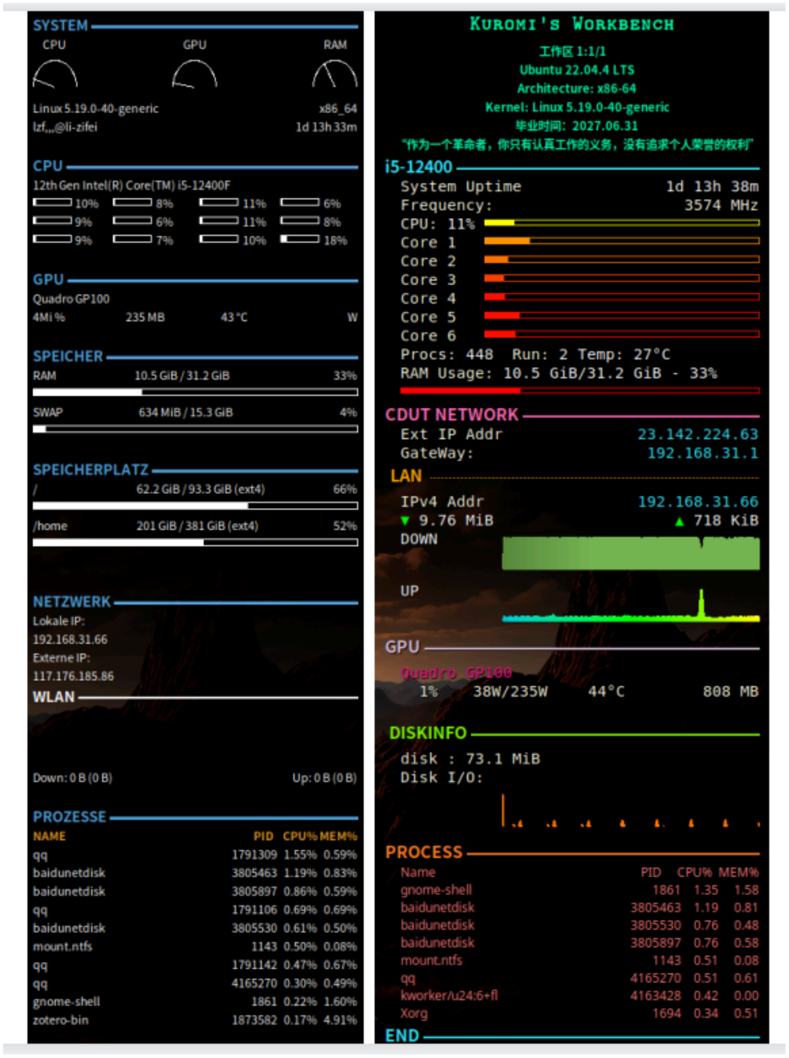
插件列表

下面是使用的部分插件，你也可以查看官方文档来自行配置插件

1. https://github.com/neoclide/coc.nvim
2. https://github.com/mhinz/vim-startify
3. https://github.com/Yggdroot/LeaderF
4. https://github.com/ap/vim-css-color
5. https://github.com/vim-airline/vim-airline
6. https://github.com/vim-airline/vim-airline-themes
7. https://github.com/gcmt/wildfire.vim
8. https://github.com/tpope/vim-surround
9. https://github.com/justinmk/vim-sneak

Conky

- 什么是Conky
  Conky 是一个能够在桌面上，以文本或图形显示各种数据的软件，类似于一个监控看板。显示的数据可以是系统时间，CPU 或内存的使用情况；也可以是网络传输过来的天气信息。功能可以说相当强大。效果图：



```
SYSTEM
CPU                     GPU                     RAM
Linux 5.19.0-40-generic                      x86_64
lzf,,,@li-zifei                           1d 13h 33m

CPU
12th Gen Intel(R) Core(TM) i5-12400F
10%     8%      11%     6%
9%      6%      11%     8%
9%      7%      10%     18%

GPU
Quadro GP100
4Mi %       235 MB      43 °C            W

SPEICHER
RAM          10.5 GiB / 31.2 GiB          33%
SWAP          634 MiB / 15.3 GiB           4%

SPEICHERPLATZ
/            62.2 GiB / 93.3 GiB (ext4)   66%
/home        201 GiB / 381 GiB (ext4)     52%

NETZWERK
Lokale IP:
192.168.31.66
Externe IP:
117.176.185.86
WLAN

Down: 0 B (0 B)                  Up: 0 B (0 B)

PROZESSE
NAME                  PID  CPU%  MEM%
qq                1791309  1.55% 0.59%
baidunetdisk      3805463  1.19% 0.83%
baidunetdisk      3805897  0.86% 0.59%
qq                1791106  0.69% 0.69%
baidunetdisk      3805530  0.61% 0.50%
mount.ntfs           1143  0.50% 0.08%
qq                1791142  0.47% 0.67%
qq                4165270  0.30% 0.49%
gnome-shell          1861  0.22% 1.60%
zotero-bin        1873582  0.17% 4.91%
```

```
KUROMI'S WORKBENCH
工作区 1:1/1
Ubuntu 22.04.4 LTS
Architecture: x86-64
Kernel: Linux 5.19.0-40-generic
毕业时间：2027.06.31
"作为一个革命者，你只有认真工作的义务，没有追求个人荣誉的权利"
i5-12400
  System Uptime                   1d 13h 38m
  Frequency:                        3574 MHz
  CPU: 11%
  Core 1
  Core 2
  Core 3
  Core 4
  Core 5
  Core 6
  Procs: 448  Run: 2 Temp: 27°C
  RAM Usage: 10.5 GiB/31.2 GiB - 33%
CDUT NETWORK
  Ext IP Addr                  23.142.224.63
  GateWay:                      192.168.31.1
 LAN
  IPv4 Addr                    192.168.31.66
  ▼ 9.76 MiB                       ▲ 718 KiB
  DOWN

  UP
GPU
  Quadro GP100
    1%    38W/235W    44°C            808 MB
 DISKINFO
  disk : 73.1 MiB
  Disk I/O:
PROCESS
  Name                 PID   CPU%  MEM%
  gnome-shell          1861   1.35  1.58
  baidunetdisk      3805463   1.19  0.81
  baidunetdisk      3805530   0.76  0.48
  baidunetdisk      3805897   0.76  0.58
  mount.ntfs           1143   0.51  0.08
  qq                4165270   0.51  0.61
  kworker/u24:6+fl  4163428   0.42  0.00
  Xorg                 1694   0.34  0.51
END
```

1. Conky安装

   ```
   sudo apt install conky
   sudo mkdir /home/lzf/softwares/conky
   sudo nvim /home/lzf/softwares/.conkyrc
   ```

2. 配置Conkyrc文件

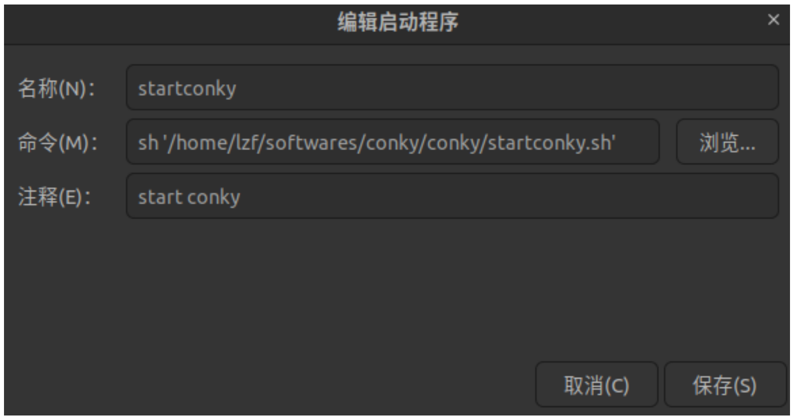
```
conky.config = {
    alignment = 'top_right',
    background = false,
    border_width = 0.1,
    cpu_avg_samples = 4,
    default_color = 'white',
    default_outline_color = 'gray',
    default_shade_color = 'black',
    draw_borders = true,
    draw_graph_borders = false,
    draw_outline = false,
    draw_shades = yes,
    use_xft = true,
    font = 'DejaVu Sans Mono:size=11',
    gap_x = 10,                                 #窗口位置
    gap_y = 40,
    minimum_height = 5,
    minimum_width = 5,
    net_avg_samples = 2,
    double_buffer = true,
    out_to_console = false,
    out_to_stderr = false,
    extra_newline = false,
    own_window = true,
    own_window_colour = '000000',
    own_window_class = 'Conky',
    own_window_argb_visual = true,
    own_window_type = 'dock',
    own_window_transparent = true,
    own_window_hints = 'undecorated,below,sticky,skip_taskbar,skip_pager',
    stippled_borders = 0,
    update_interval = 1,
    uppercase = false,
    use_spacer = 'none',
    show_graph_scale = false,
    show_graph_range = false
}
conky.text = [[
#${image ~/.face -p 180,5 -s 70x70 -f 86400}${image ~/.pacman -p 20,9 -s 60x60 -f 86400}
${font Latin Modern Mono Caps:bold:size=14}${alignc}${color 00ffae}Kuromi's Workbench
${font Entopia:bold:size=8.5}${alignc}${desktop_name}:${desktop}/$desktop_number
${font Entopia:bold:size=8.5}${alignc}   ${exec hostnamectl | grep System | cut -c19-40}
${font Entopia:bold:size=8.5}${alignc}   ${exec hostnamectl | grep Architecture | cut -c5-30}
${font Entopia:bold:size=8.5}${alignc}   ${exec hostnamectl | grep Kernel | cut -c11-47}
${font Entopia:bold:size=8.5}${alignc}   毕业时间：2027.06.31
${font Entopia:bold:size=8.5}${alignc}   “作为一个革命者，你只有认真工作的义务，没有追求个人荣誉的权利”
${font Entopia:bold:size=12}${color 33E9FF}i5-12400 ${hr 2}${font}
${offset 15}${color FFFDE2}System Uptime ${alignr}$color $uptime
${offset 15}${color FFFDE2}Frequency: ${alignr}${freq dyn} MHz
${offset 15}${color FFFDE2}CPU:$color ${cpu}% ${color yellow}${cpubar 5}${color FFFDE2}
${offset 15}Core 1   ${color ff9300}${cpubar cpu1 6}${color FFFDE2}
${offset 15}Core 2   ${color ff7300}${cpubar cpu2 6}${color FFFDE2}
${offset 15}Core 3   ${color ff4300}${cpubar cpu3 6}${color FFFDE2}
${offset 15}Core 4   ${color ff1300}${cpubar cpu4 6}${color FFFDE2}
${offset 15}Core 5   ${color ff1300}${cpubar cpu5 6}${color FFFDE2}
${offset 15}Core 6   ${color ff1300}${cpubar cpu6 6}${color FFFDE2}
${offset 15}${font}${color FFFDE2}Procs:$color $processes  ${color FFFDE2}Run:$color $running_processes Temp: ${acpitemp}°C
${offset 15}${color FFFDE2}RAM Usage:$color $mem${color0}/${color4}$memmax - $memperc%
${offset 15}${color FF0000}${membar 5}
${font Entopia:bold:size=12}${color FF69B4}CDUT NETWORK ${hr 2}${font}
${offset 15}${color FFFDE2}Ext IP Addr ${color 33E9FF}${alignr}${exec cat /home/lzf/env/myip.txt}
${offset 15}${color FFFDE2}GateWay:${color 33E9FF}${alignr}${gw_ip}
${offset 5}${font Entopia:bold:size=12}${color orange}LAN  ${stippled_hr 1}
${offset 15}${font}${color FFFDE2}IPv4 Addr ${color 33E9FF}${alignr}${addr eno1}
${offset 15}${color green}${font}▼ $color${downspeed eno1} ${alignr}${color green}▲ $color${upspeed eno1}
${offset 15}${font}${color}DOWN ${downspeedgraph eno1 32,0 324D23 77B753}
${offset 15}${font}${color}UP   ${upspeedgraph eno1 32,0 104E8B ffff00}
${font Entopia:bold:size=12}${color D8BFD8}GPU ${hr 2}
${offset 15}${font}${color FF1493}{exec nvidia-smi | grep % | cut -c 85-92}
${offset 15}${font}${color FFFDE2}${exec nvidia-smi | grep % | cut -c 71-73}%${goto 80}${exec nvidia-smi | grep % | cut -c 31-33}W/${exec nvidi

${font Entopia:bold:size=12}${color 7cfc00} DISKINFO ${hr 2}
${offset 15}${font}${color FFFDE2}disk : ${diskio}
${offset 15}${font}${color FFFDE2}Disk I/O:
${offset 15}${font}${diskiograph 32,0 ff7300 ff7300}
${font Entopia:bold:size=12}${color FF7F24}PROCESS ${hr 2}${font}
${offset 15}${font Noto sans:size=9}${color FF7878}Name ${alignr}PID     CPU%  MEM%
${offset 15}${color FF7878}${top name 1} ${alignr}${top pid 1}  ${top cpu 1}   ${top mem 1}
${offset 15}${color FF7878}${top name 2} ${alignr}${top pid 2}  ${top cpu 2}   ${top mem 2}
${offset 15}${color FF7878}${top name 3} ${alignr}${top pid 3}  ${top cpu 3}   ${top mem 3}
${offset 15}${color FF7878}${top name 4} ${alignr}${top pid 4}  ${top cpu 4}   ${top mem 4}
${offset 15}${color FF7878}${top name 5} ${alignr}${top pid 5}  ${top cpu 5}   ${top mem 5}
${offset 15}${color FF7878}${top name 6} ${alignr}${top pid 6}  ${top cpu 6}   ${top mem 6}
${offset 15}${color FF7878}${top name 7} ${alignr}${top pid 7}  ${top cpu 7}   ${top mem 7}
${offset 15}${color FF7878}${top name 8} ${alignr}${top pid 8}  ${top cpu 8}   ${top mem 8}
${font Entopia:bold:size=12}${color 33E9FF}END ${hr 2}${font}
]]
```

上述conky文件对应图中第二张图的Style
查看效果： `conky -c ~/.config/conky/.conkyrc`

- 设置开机启动
  设置一个脚本
  `sudo nvim /home/lzf/softwares/conky/startconky.sh`
  写入如下内容：

```
sleep 5
conky -c /home/lzf/softwares/conky/.conkyrc
```

在应用列表中找到startup application，然后加入以下条目：

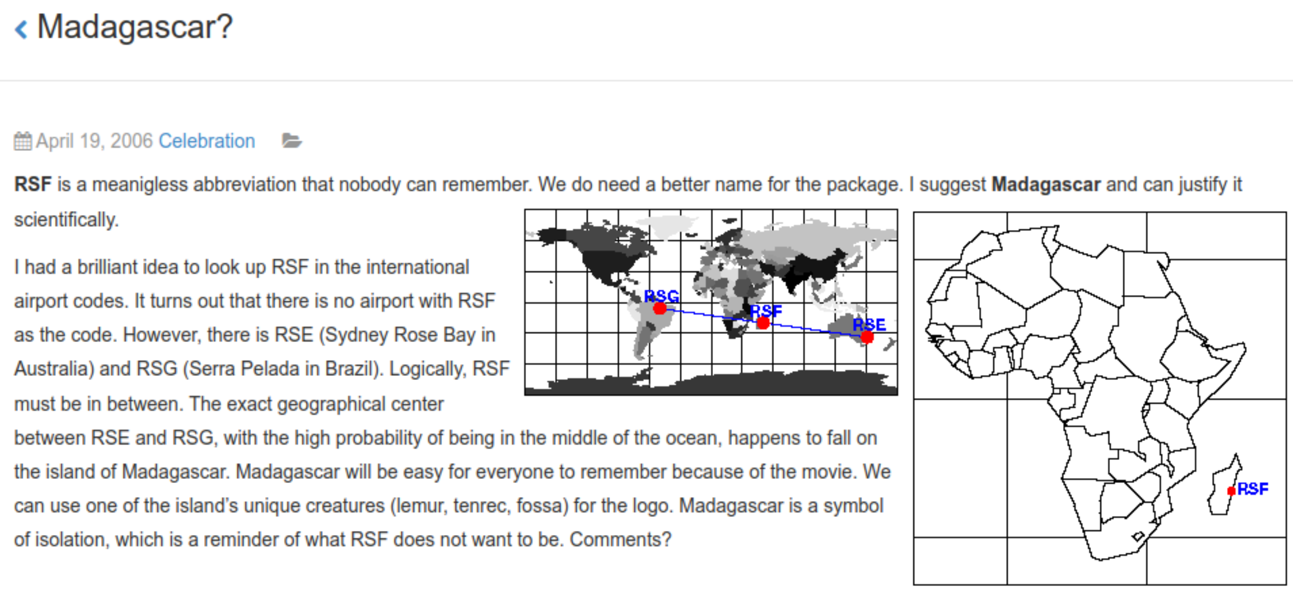


名称(N)： startconky
命令(M)： sh '/home/lzf/softwares/conky/conky/startconky.sh'
注释(E)： start conky

Chapter2:地震软件篇

Madagascar

为什么要叫mada（



‹ Madagascar?

April 19, 2006 Celebration

RSF is a meanigless abbreviation that nobody can remember. We do need a better name for the package. I suggest **Madagascar** and can justify it scientifically.

I had a brilliant idea to look up RSF in the international airport codes. It turns out that there is no airport with RSF as the code. However, there is RSE (Sydney Rose Bay in Australia) and RSG (Serra Pelada in Brazil). Logically, RSF must be in between. The exact geographical center between RSE and RSG, with the high probability of being in the middle of the ocean, happens to fall on the island of Madagascar. Madagascar will be easy for everyone to remember because of the movie. We can use one of the island's unique creatures (lemur, tenrec, fossa) for the logo. Madagascar is a symbol of isolation, which is a reminder of what RSF does not want to be. Comments?

Madagascar是一个用于多维数据分析和可重复计算实验的开源软件包。它的使命是提供方便而强大的环境、简单的地震数据常用处理函数、和便捷的代码专业工具。适用于在地球物理学和相关领域从事数字图像和数据处理的研究人员。使用mada项目管理系统开发的技术以历史记录的形式传输，这些历史成为“计算配方”，由系统用户进行验证、交换和修改。

安装mada

1. 下载mada安装文件:
   `git clone https://github.com/ahay/src RSFSRC` 或者 `svn co https://github.com/ahay/src/trunk RSFSRC`

- 预安装：
  不同系统需要安装不同的依赖软件
  为了方便查错所以这里分开写供安装使用吧
  - `sudo apt-get install libxaw7-dev freeglut3-dev libnetpbm10-dev libgd-dev`
  - `sudo apt-get install libplplot-dev libavcodec-dev libcairo2-dev libjpeg-dev`
  - `sudo apt-get install swig python3-dev python3-numpy g++ gfortran`
  - `sudo apt-get install libopenmpi-dev libfftw3-dev libsuitesparse-dev scons git`

2. 配置安装路径

- 配置matlab环境变量,制作mex链接
  ```
  sudo ln -s /home/lzf/softwares/Mat-lab/bin/mex /usr/local/bin/mex
  ```
- 配置Madagascar安装环境和api接口
  ```
  ./configure --prefix= /directory/where/you/want/madagascar/installed/RSFROOT
  ```
  您可以通过运行 scons -h 来获取可自定义变量的完整列表。例如，要安装 matlab API 绑定以及基本软件包，请运行：
  `API=matlab` *
  `CUDA TOOLKIT PATH=` *
  其中prefix 指定安装路径,API和CUDA设置根据CUDA路径以及matlab安装路径来决定
  该步骤有时会报错如下，此时检查config.py文件，如果正常生成并且里面的设置正确，就可直接执行安装了。

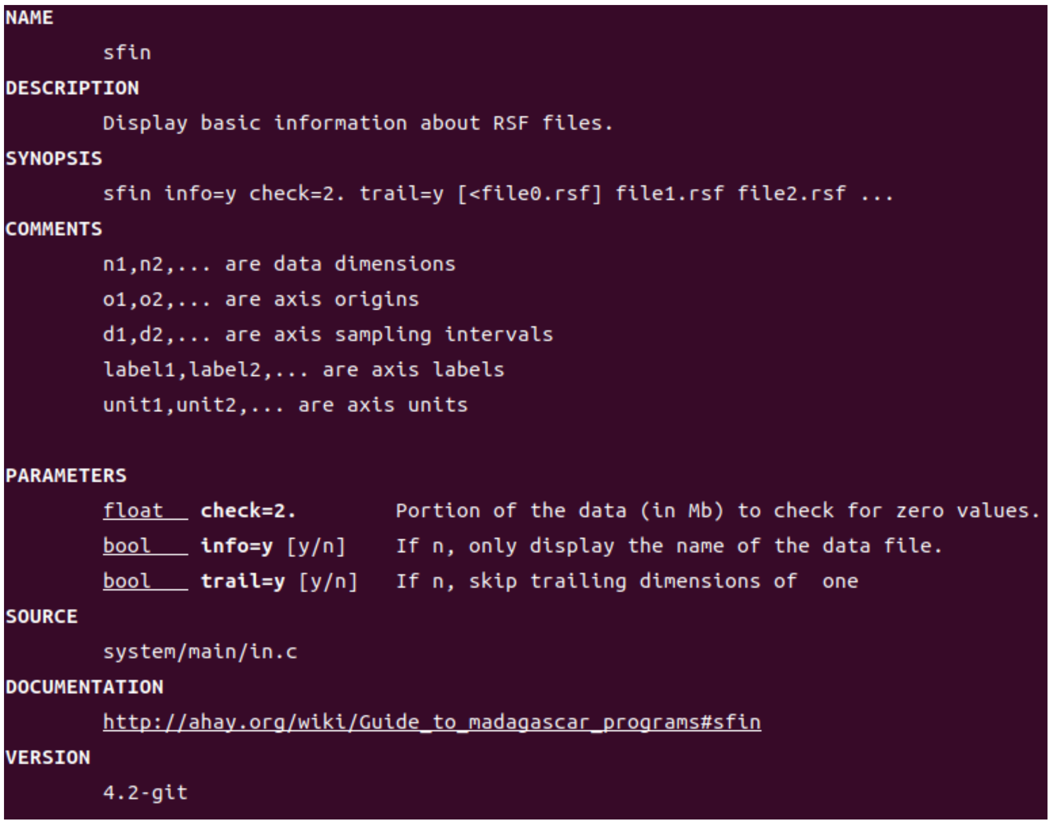
```
scons: Reading SConscript files ...
scons: done reading SConscript files.
usage: scons [OPTION] [TARGET] ...

SCons Error: no such option: --prefix
------------------------
Done with configuration.
```

3. 安装
   `scons install` 或者 `make install`
4. 配置环境变量
   以下内容写入.rc环境文件中:
   - `export RSFROOT=/home/lzf/softwares/Madagascar/RSFROOT` 马达安装路径
   - `source $RSFROOT/share/madagascar/etc/env.sh` 加载马达环境变量
   - `export DATAPATH=/home/lzf/data/.Mada/data/` 马达二进制数据和rsf文件是分离开的，该命令设置马达二进制数据保存路径
   - `export RSFFIGS=/home/lzf/data/.Mada/Figure/` 马达生成文章是保存的路径
   - `export RSFALTFIGS=/home/lzf/data/.Mada/Figure1/` 马达test例子存放位置
   - `export RSFMEMSIZE=30000` 允许最大的内存(Mb)

   代码块如下:

```
export RSFROOT=/home/lzf/softwares/Madagascar/RSFROOT
source $RSFROOT/share/madagascar/etc/env.sh
export DATAPATH=/home/lzf/data/.Mada/data/
export RSFFIGS=/home/lzf/data/.Mada/Figure/
export RSFALTFIGS=/home/lzf/data/.Mada/Figure1/
export RSFMEMSIZE=30000
```

5. 安装成功测试
   终端输入 `sfin`:



```
NAME
        sfin
DESCRIPTION
        Display basic information about RSF files.
SYNOPSIS
        sfin info=y check=2. trail=y [<file0.rsf] file1.rsf file2.rsf ...
COMMENTS
        n1,n2,... are data dimensions
        o1,o2,... are axis origins
        d1,d2,... are axis sampling intervals
        label1,label2,... are axis labels
        unit1,unit2,... are axis units

PARAMETERS
        float   check=2.        Portion of the data (in Mb) to check for zero values.
        bool    info=y [y/n]    If n, only display the name of the data file.
        bool    trail=y [y/n]   If n, skip trailing dimensions of  one
SOURCE
        system/main/in.c
DOCUMENTATION
        http://ahay.org/wiki/Guide_to_madagascar_programs#sfin
VERSION
        4.2-git
```

下面是一个简单的 SConstruct 文件:

```
#
# Setting up
#
from rsf.proj import *

#
# Make filter.rsf
#
Flow('filter',None,'spike n1=1000 k1=300 | bandpass fhi=2 phase=y')

#
# Make filter.vpl
#
Result('filter','wiggle clip=0.02 title="Welcome to Madagascar"')
End()
```

6. 英文版手册

```
======================
Madagascar Installation
======================

Prerequisites
=============

   1. C compiler. ANSI-compliant compiler such as GCC should work. GCC
   usually comes pre-installed on Linux machines.

   2. Python interpreter. Python is an interpretable programming
   language. It is used in Madagascar installation scripts and project
   management scripts. Python comes pre-installed on some
   platforms. Madagascar supports both Python 2.7 and Python 3.

For more information see:
http://ahay.org/wiki/Main_Page
http://ahay.org/wiki/Installation
http://ahay.org/wiki/Advanced_Installation

Software construction
=====================

   1. Configuration.

      Change to the top Madagascar source directory and run

      ./configure --prefix=/directory/where/you/want/madagascar/installed

      You can examine the config.py file that this command
      generates. Additional options are available. You can obtain a
      full list of customizable variables by running "scons -h". For
      example, to install Fortran-90 API bindings in addition to the
      basic package, run

      ./configure --prefix=/directory/where/you/want/madagascar/installed \
      API=fortran-90

   2. Building and installing the package.

      Run "make install" or the following two commands in succession:

      make
      make install

      If you need "root" privileges for installing under $RSFROOT, you
      may need to run

      make
      su
      make install

      or

      make
      sudo make install

   3. User setup

      If your shell is sh or bash, add to your $HOME/.bashrc and
      $HOME/.bash_profile files the line

      source RSFROOT/share/madagascar/etc/env.sh
```

where RSFROOT is the install directory you specified in the --prefix option to ./configure. If your shell is (t)csh, add to your $HOME/.cshrc file the line

```
source RSFROOT/share/madagascar/etc/env.csh
```

Be aware that on some systems the default value for DATAPATH set in the script above may get automatically cleaned at some intervals, so if you want to keep your data binaries for a long time, set DATAPATH in your resource file to another location where you have write access and that allows large files. Remember that the value of DATAPATH should have a slash at the end.

Testing Your Installation

Here are a few simple tests and and a brief introduction to Madagascar:

Typing any Madagascar command in a terminal window without parameters should generate a brief documentation on that command. Try one of the following:

```
sfin
sfattr
sfspike
sfbandpass
sfwiggle
```

If you get an error like "Command not found", you may not have your PATH environment variable set correctly, or you may need to issue the rehash command.

Now try making a simple Madagascar data file:

```
sfspike n1=1000 k1=300 > spike.rsf
```

This command generates a one dimensional list of 1000 numbers, all zero except for a spike equal to one at position 300. If this generates an error like

```
Cannot write to data file /path/spike.rsf@: Bad file descriptor
```

you may need to create the directory pointed to by your DATAPATH environment variable.

The file spike.rsf is a text header. The actual data are stored in the binary file pointed to by the in parameter in the header. You can look at the header file directly with more, or better, examine the file properties with

```
sfin spike.rsf
```

You can learn more about the contents of spike.rsf with

```
sfattr < spike.rsf
```

The following command applies a bandpass filter to spike.rsf and puts the result in filter.rsf:

```
sfbandpass fhi=2 phase=y < spike.rsf > filter.rsf
```

The following command makes a graphics file from filter.rsf:

```
sfwiggle clip=0.02 title="Welcome to Madagascar" < filter.rsf > filter.vpl
```

If you have an X11 display program running, and your DISPLAY
environment variable is set correctly, you can display the graphics file with:

```
        sfpen < filter.vpl
```

You can pipe Madagascar commands together and do the whole thing at once like
this:

```
        sfspike n1=1000 k1=300 | sfbandpass fhi=2 phase=y | \
        sfwiggle clip=0.02 title="Welcome to Madagascar" | sfpen
```

If you have SCons installed, you can use it to automate Madagascar processing.
Here is a simple SConstruct file to make filter.rsf and filter.vpl :

```
####################################
#
# Setting up
#
from rsf.proj import *

#
# Make filter.rsf
#
Flow('filter',None, 'spike n1=1000 k1=300 | bandpass fhi=2 phase=y')

#
# Make filter.vpl
#
Result('filter','wiggle clip=0.02 title="Welcome to Madagascar"')

End()
####################################
```

Put the file in an empty directory, give it the name SConstruct,
cd to that directory, and issue the command:

```
        scons
```

The graphics file is now stored in the Fig subdirectory. You can
view it manually with:

```
        sfpen Fig/filter.vpl
```

... or you can use:

```
        scons view
```

When an SConstruct file makes more than one graphics file, the

```
        scons view
```

command will display all of them in sequence.

Now edit the SConstruct file: change the title string on the
Result line to "Hello World!", save the file, and rerun the scons
command.

You will see that scons has figured out that the file
filter.rsf does not need to be rebuilt because nothing that affects
it has changed. Only the file filter.vpl is rebuilt.

Bugs

```
Please report all problems encountered during software construction to
the RSF-user mailing list:

https://lists.sourceforge.net/lists/listinfo/rsf-user

You can also send suggestions for improvement of this document to the list.
```

mada 保存图片：

```
vpconvert *.vpl format=jpg color=y bgcolor=white
```

mada常用命令:

sfadd减法：`add scale=1,-1 ${SOURCES[1]}`
sfwindow参数：n#=* 指的在第#个道集采多* 长，f#=* 指的是采样间隔，

Seismic Unix

SU是科罗拉多州矿业学院开发的一个免费地震处理软件。国内外很多科研人员及学生都借助于他来进行创作，SU开放源代码，可以方便地在其基础上进行再创作。其实有了mada就不用su了，一个爹生的。

安装su

1. 下载su安装文件:
   https://nextcloud.seismic-unix.org/s/LZpzc8jMzbWG9BZ

- 解压

```
mkdir /where/you/want/su/put
cd /where/you/want/su/put
gunzip cwp_su_all_xx_tar.gz
tar -xvf cwp_su_all_xx_tar
```

- 预安装：
  不同系统需要安装不同的依赖软件
  为了方便查错所以这里同样分开写供安装使用
  - sudo apt-get install build-essential
  - sudo apt-get install libx11-dev
  - sudo apt-get install libxt-dev
  - sudo apt-get install freeglut3-dev
  - sudo apt-get install libxmu-dev
  - sudo apt-get install libxi-dev
  - sudo apt-get install gfortran

2. 配置环境

- .rc环境变量文件中写入

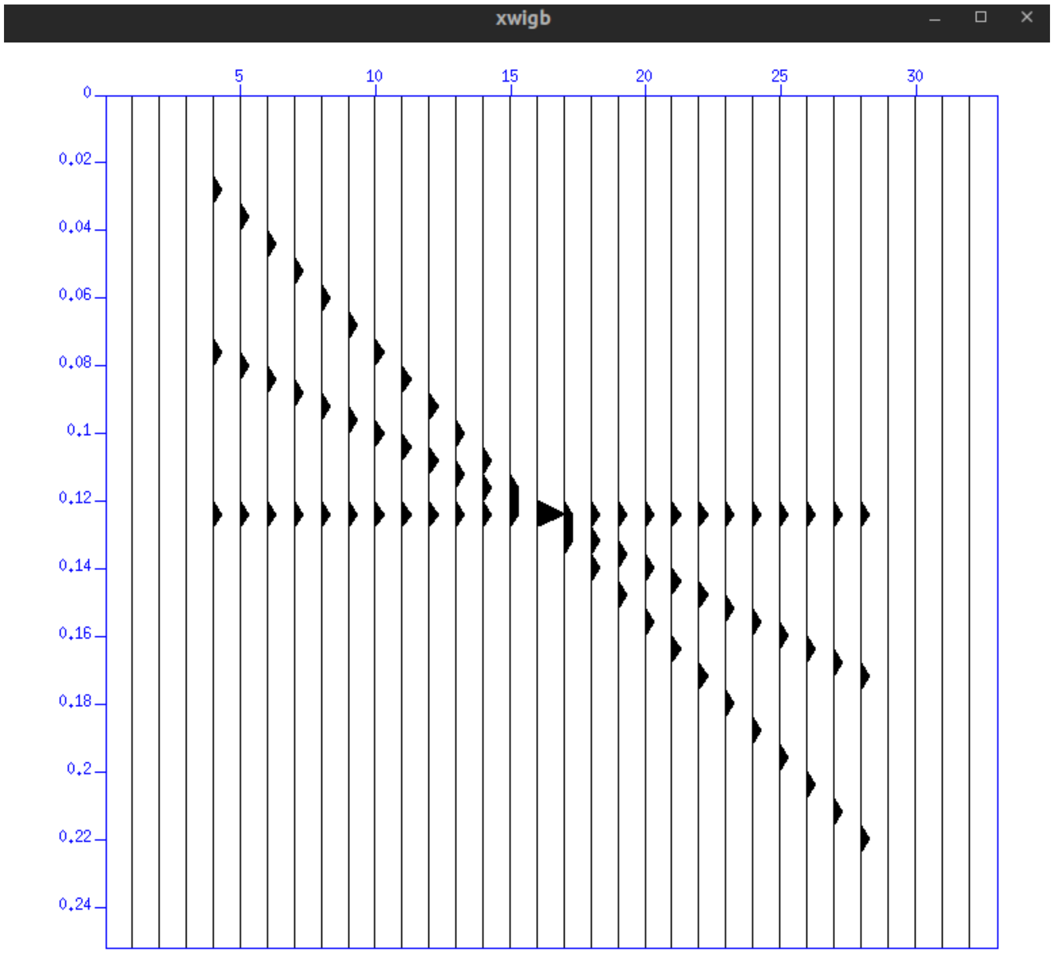
```
export CWPROOT=/where/you/su installpack/su
export PATH=$PATH:/where/you/su installpack/su/bin
```

强烈不推荐任何直接替换Makefile.config的博文方法，如需特定的Makefile.config格式请打开su中的configs文件夹结合自身电脑选择并替换，注意su部分版本会因为Linux内核版本而安装失败。

- 安装

```
cd $CWPROOT/src
make install
make xtinstall
make finstall # fortran模块
make mglinstall
make utils
make xminstall
make sfinstall  # segd模块
```

3. 安装成功测试
   终端输入 `suplane | suxwigb`



二进制数据绘图

```
ximage <acc_vp_2.dat n1=400 perc=99 cmap=rgb2
```

GeoEast

软件开发模块

1. Geoeast自带了常用的许多python库，使用深度学习训练好的网络模型在进行封装的时候只需要把代码调整好只使用GPU运行即可。

Chapter3:编程语言篇

C/C++/C-cuda/mpich

- 错误调试
  在makefile文件中添加 `CFALGS= -g -Wall` 生成调试文件
  具体做法：
  `$(MPICC) -g -Wall -g -c`
  `$(NVCC)  -g -w -c`
  然后在 GDB 中运行程序并进行调试。例如，你可以使用 run 命令来运行程序，使用 break 命令设置断点，使用 print 命令打印变量的值，等等。详细的使用方法可以参考 GDB 的文档或者相关教程。
- mpicc
  MPICC运行结果不正确大部分原因是进程问题，可以将 `mpicc -np` 参数改为1然后尝试运行
- makefile模版
  根据需要更改对应路径位置

```
# ==================================================================================
#    Copyright (C) 2024 Chengdu University of Technology.
#    Copyright (C) 2024 Zifei Li.
#
#    Filename: Makefile
#    Author: Zifei Li
#    Institute: Chengdu University of Technology
#    Email: 202005050218@stu.cdut.edu.cn
#    Work: 2024/02/04/
#    Function:
#
#    This program is free software: you can redistribute it and/or modify it
#    under the terms of the GNU General Public License as published by the Free
#    Software Foundation, either version 3 of the License, or an later version.
#=================================================================================
#!/bin/bash
#include /home/lzf/softwares/makeopt/makefile.opt`
CXX = g++
CC = gcc
###################### fftw #############################
#FFTW_DIR = /home/lzf/softwares/FFTW
#FFTW_INC = -I$(FFTW_DIR)/include
#FFTW_LIB = -I$(FFTW_DIR)/lib
#################### Madagascar ########################
#${RSFROOT}= /home/lzf/softwares/madagascar
#RSFROOT_lib = -L${RSFROOT}/lib -lrsf -lrsf++ -lm
#################### cuda ########################
#CUDA_HOME = /usr/local/cuda-12.2
#NVCC = $(CUDA_HOME)/bin/nvcc
#################### MPICH ########################
#MPICC_HOME = /home/lzf/softwares/MPICH
#MPICC = $(MPICC_HOME)/bin/mpicc

MYPROGS=./myprogs
CFILAGS = -I$(MYPROGS)
EXECNAME = Obser

LINK = -fPIC -lm
CFILES =  .c   .cpp
OBJECTS = .o

all:
  $(CC) -w -c $(CFILES)   $() $(LINK)
  $(CC) -o $(EXECNAME)    $() $(LINK)
  ./Obser
#     mpirun -np 5./Obser
clean:
    rm -f *.o Obser
```

一些环境配置

```
# >>> matlab <<<
export JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64
export PATH=$PATH:/home/lzf/softwares/matlab/bin
alias mrun="matlab -nodesktop -nosplash -logfile `date +%Y_%m_%d-%H_%M_%S`.log -r"

# >>> MADAGASCAR <<<
source ~/softwares/Madagascar/RSFROOT/share/madagascar/etc/env.sh
export DATAPATH=/media/lzf/Work/data/RSFDATA/data/
export RSFFIGS=/media/lzf/Work/data/RSFDATA/Figure/
export RSFALTFIGS=/media/lzf/Work/data/RSFDATA/Figure1/
export RSFMEMSIZE=30000

# >>> Seismic Unix <<<
export CWPROOT_HOME=/home/lzf/softwares/seismic_unix/cwp
export PATH=$PATH:${CWPROOT_HOME}/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${CWPROOT_HOME}/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${CWPROOT_HOME}/include

# >>> MPICH <<<
export MPI_HOME=/home/lzf/softwares/MPICH/MPICC
export PATH=$PATH:${MPI_HOME}/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${MPI_HOME}/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${MPI_HOME}/include

# >>> cuda env <<<
export CUDA_HOME=/usr/local/cuda/bin
export PATH=$PATH:${CUDA_HOME}/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${CUDA_HOME}/lib64
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${CUDA_HOME}/include

# >>> FFTW env <<<
export FFTW_DIR=/home/lzf/softwares/FFTW
export FFTW=/home/lzf/softwares/FFTW/install_pack/fftw-2.1.5
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${FFTW_DIR}/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${FFTW_DIR}/include

# >>> curveLab env <<<
export FDCT=/home/lzf/softwares/CurveLab/CurveLab-2.1.3

# >>> SWIG <<<
export SWIG_HOME=/home/lzf/softwares/SWIG
export PATH=$PATH:${SWIG_HOME}/bin

# >>> texlive env <<<
export TexMan="/usr/local/texlive/2023/texmf-dist/doc/man"
export TexInfo="/usr/local/texlive/2023//texmf-dist/doc/info"
export TexLive="/usr/local/texlive/2023/bin/x86_64-linux"
export MANPATH="$MANPATH:$TexMan"
export INFOPATH="$INFOPATH:$TexInfo"
export PATH="$PATH:$TexLive"
```

matlab

终端运行

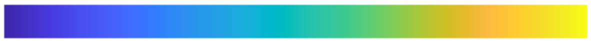
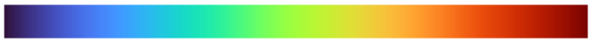
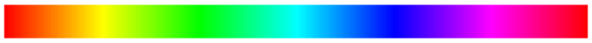
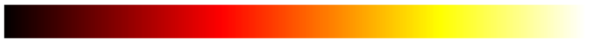
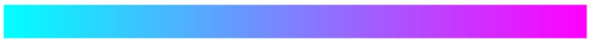
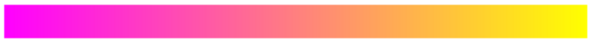
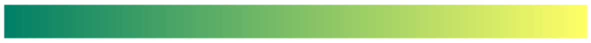
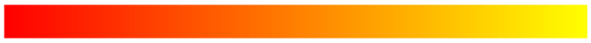
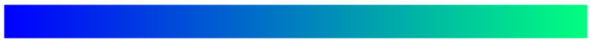
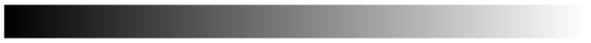
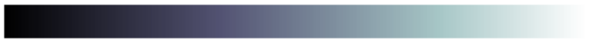
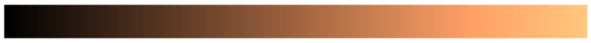
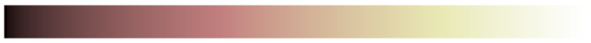
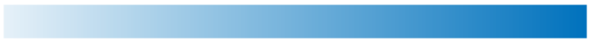
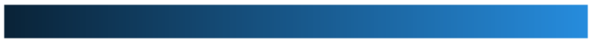
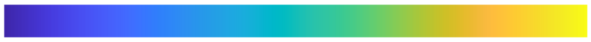
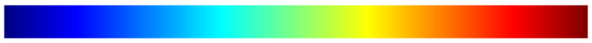
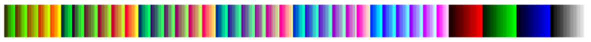
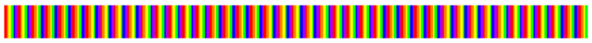
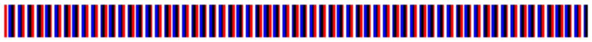
```
matlab -batch filename(no'.m'!!) taskset -c 1-16(core_number 1-16)
```

绘图demo

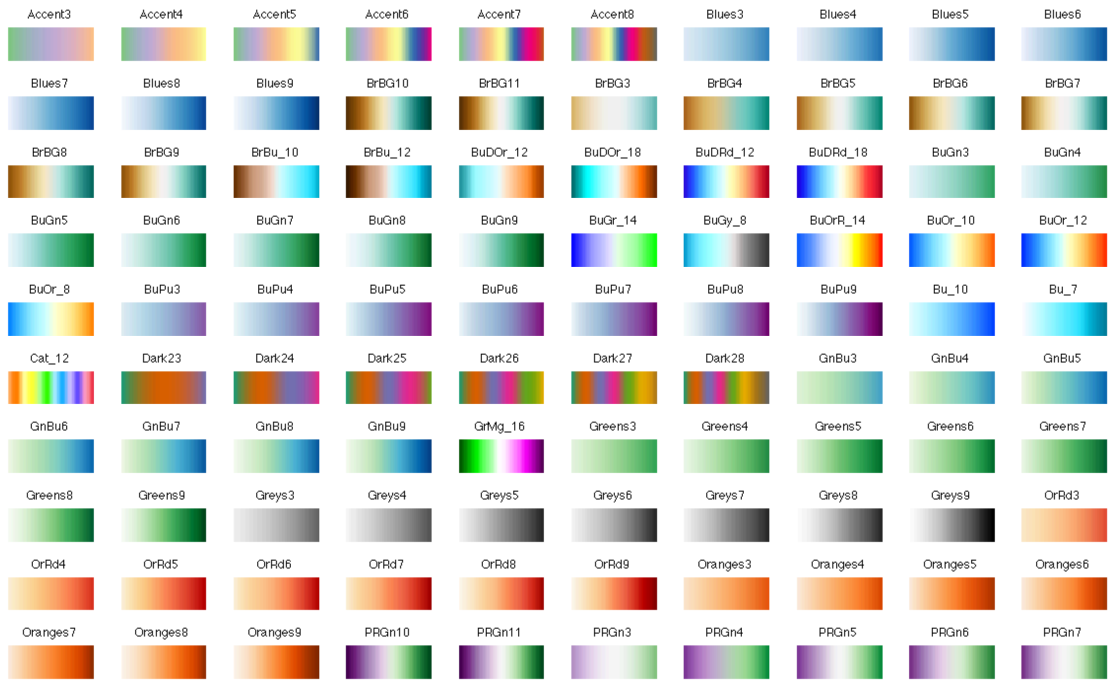
```
p = 0;p = p+1;figure(p);
clip =
mm=
filename2 = ['11source_seismogram_obs_sing_csg.dat'];
wave = zread(filename2,[n1,n2]);
imagesc(wave1,[-clip,clip]);colormap(mm);
%axis off
set(gca,'looseInset',[0 0 0 0]);
set(gcf,'unit','normalized','position',[0.1,0.1,0.3,0.8] );
```

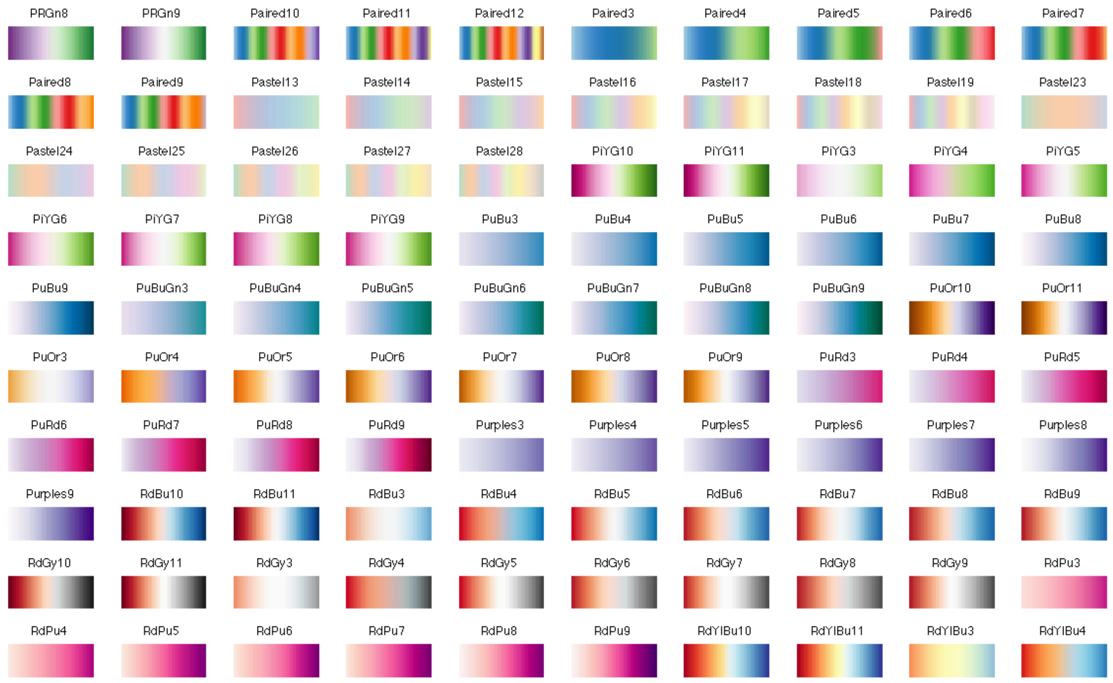
绘图色块

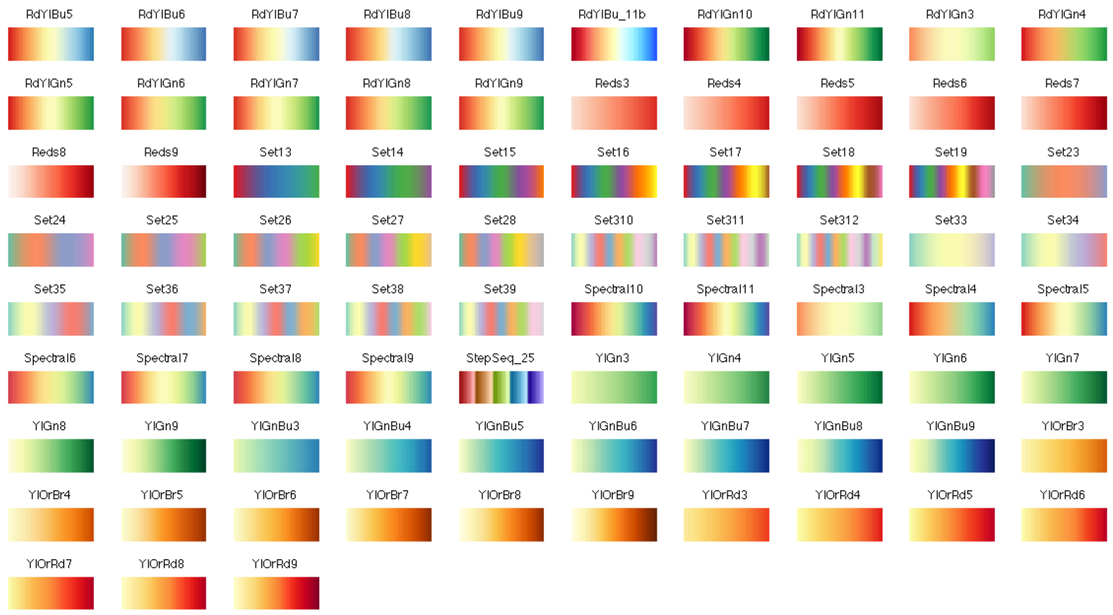
- 自带色块：

colormap()	color
parula	
turbo	
hsv	
hot	
cool	
spring	
summer	
autumn	
winter	
gray	
bone	
copper	
pink	
sky (自 R2023a 起)	
abyss (自 R2023b 起)	
jet	
lines	
colorcube	
prism	
flag	
white	

- othercolors：
  下载链接


Accent3
Accent4
Accent5
Accent6
Accent7
Accent8
Blues3
Blues4
Blues5
Blues6
Blues7
Blues8
Blues9
BrBG10
BrBG11
BrBG3
BrBG4
BrBG5
BrBG6
BrBG7
BrBG8
BrBG9
BrBu_10
BrBu_12
BuDOr_12
BuDOr_18
BuDRd_12
BuDRd_18
BuGn3
BuGn4
BuGn5
BuGn6
BuGn7
BuGn8
BuGn9
BuGr_14
BuGy_8
BuOrR_14
BuOr_10
BuOr_12
BuOr_8
BuPu3
BuPu4
BuPu5
BuPu6
BuPu7
BuPu8
BuPu9
Bu_10
Bu_7
Cat_12
Dark23
Dark24
Dark25
Dark26
Dark27
Dark28
GnBu3
GnBu4
GnBu5
GnBu6
GnBu7
GnBu8
GnBu9
GrMg_16
Greens3
Greens4
Greens5
Greens6
Greens7
Greens8
Greens9
Greys3
Greys4
Greys5
Greys6
Greys7
Greys8
Greys9
OrRd3
OrRd4
OrRd5
OrRd6
OrRd7
OrRd8
OrRd9
Oranges3
Oranges4
Oranges5
Oranges6
Oranges7
Oranges8
Oranges9
PRGn10
PRGn11
PRGn3
PRGn4
PRGn5
PRGn6
PRGn7


PRGn8
PRGn9
Paired10
Paired11
Paired12
Paired3
Paired4
Paired5
Paired6
Paired7
Paired8
Paired9
Pastel13
Pastel14
Pastel15
Pastel16
Pastel17
Pastel18
Pastel19
Pastel23
Pastel24
Pastel25
Pastel26
Pastel27
Pastel28
PiYG10
PiYG11
PiYG3
PiYG4
PiYG5
PiYG6
PiYG7
PiYG8
PiYG9
PuBu3
PuBu4
PuBu5
PuBu6
PuBu7
PuBu8
PuBu9
PuBuGn3
PuBuGn4
PuBuGn5
PuBuGn6
PuBuGn7
PuBuGn8
PuBuGn9
PuOr10
PuOr11
PuOr3
PuOr4
PuOr5
PuOr6
PuOr7
PuOr8
PuOr9
PuRd3
PuRd4
PuRd5
PuRd6
PuRd7
PuRd8
PuRd9
Purples3
Purples4
Purples5
Purples6
Purples7
Purples8
Purples9
RdBu10
RdBu11
RdBu3
RdBu4
RdBu5
RdBu6
RdBu7
RdBu8
RdBu9
RdGy10
RdGy11
RdGy3
RdGy4
RdGy5
RdGy6
RdGy7
RdGy8
RdGy9
RdPu3
RdPu4
RdPu5
RdPu6
RdPu7
RdPu8
RdPu9
RdYlBu10
RdYlBu11
RdYlBu3
RdYlBu4



python

anaconda&&pip

- 创建虚拟环境

  `conda create -n xxxxx(名字) python=3.8`
- 删除虚拟环境

  `conda remove -n xxxxx(名字) --all`
- 删除某个包

  `conda remove package_name`
- 复制虚拟环境

  `conda create -n B --clone A`
- 导出虚拟环境

  `conda env export > environment.yaml`

  `pip list --format=freeze> requirements.txt`
- 根据导出创建

  `conda env create -f environment.yaml`

  `conda install --yes --file requirements.txt`

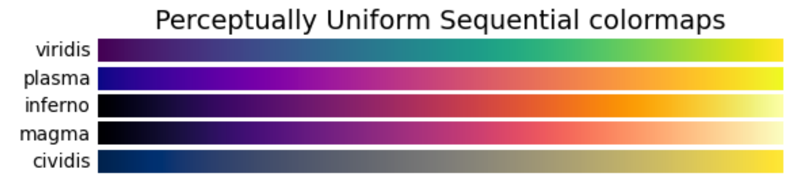
  `pip install -r requirements.txt`

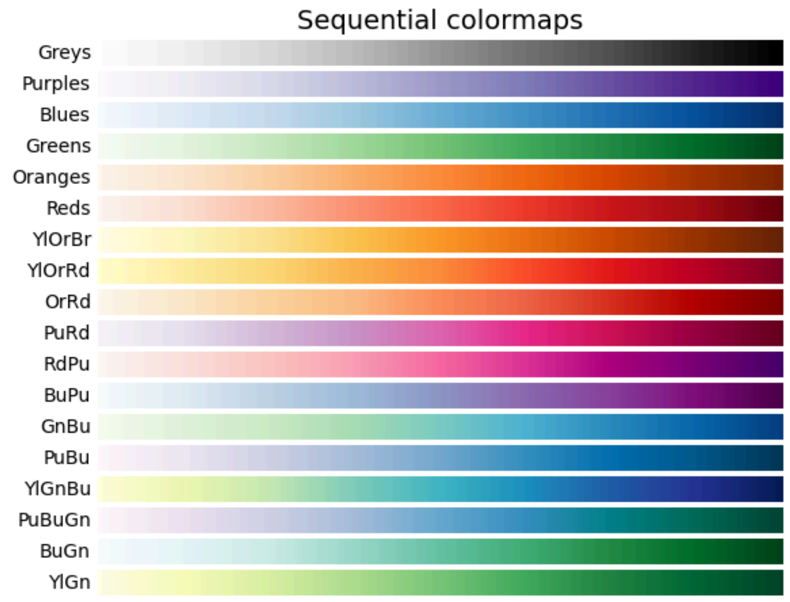
一些依赖包的install

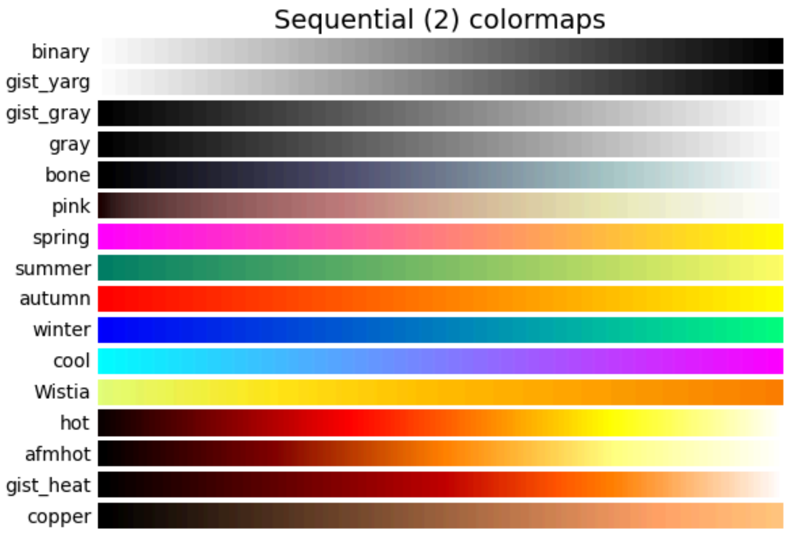
- Curvelab
  虚拟环境中安装依赖库： `python3 -m pip install git+https://github.com/PyLops/curvelops@0.23`

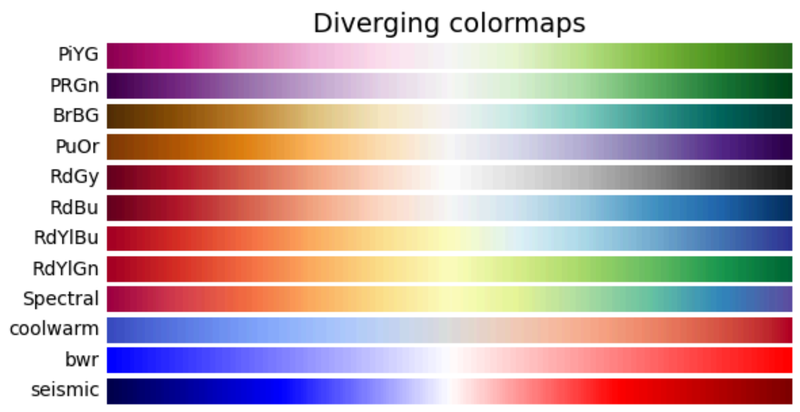
matplot 绘图色块

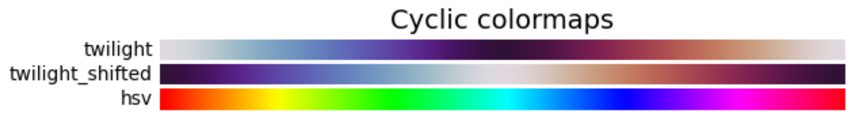
- 自带colormap

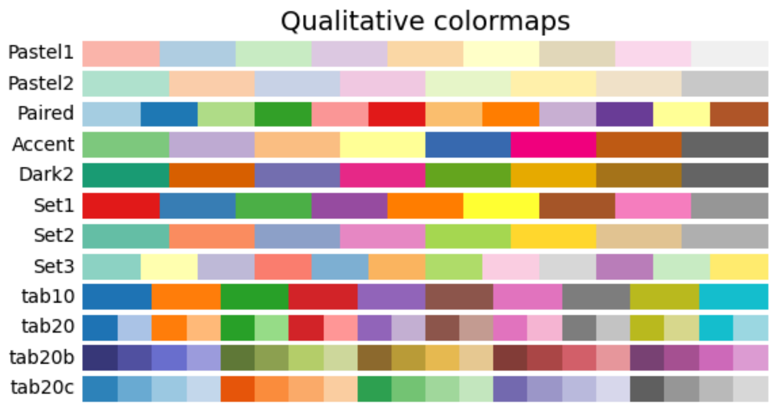

Perceptually Uniform Sequential colormaps
viridis
plasma
inferno
magma
cividis

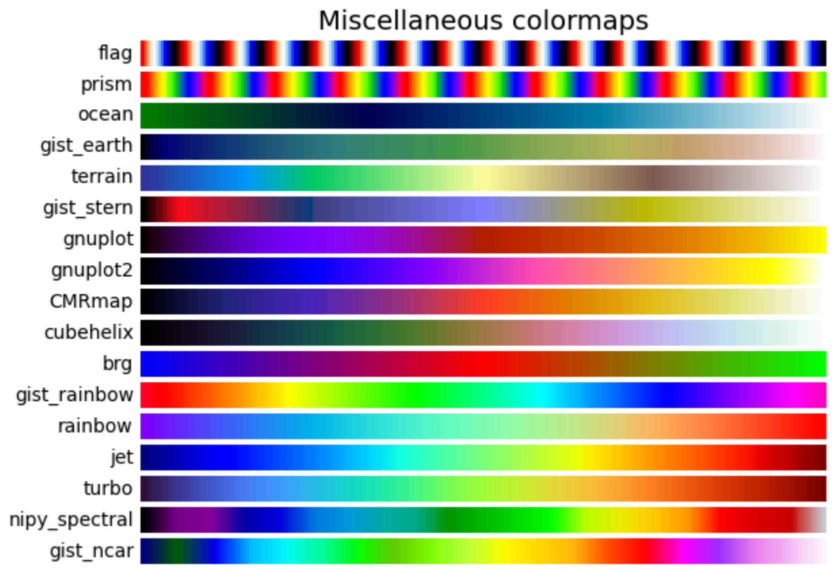

Sequential colormaps
Greys
Purples
Blues
Greens
Oranges
Reds
YlOrBr
YlOrRd
OrRd
PuRd
RdPu
BuPu
GnBu
PuBu
YlGnBu
PuBuGn
BuGn
YlGn


Sequential (2) colormaps
binary
gist_yarg
gist_gray
gray
bone
pink
spring
summer
autumn
winter
cool
Wistia
hot
afmhot
gist_heat
copper


Diverging colormaps
PiYG
PRGn
BrBG
PuOr
RdGy
RdBu
RdYlBu
RdYlGn
Spectral
coolwarm
bwr
seismic


Cyclic colormaps
twilight
twilight_shifted
hsv


Qualitative colormaps
Pastel1
Pastel2
Paired
Accent
Dark2
Set1
Set2
Set3
tab10
tab20
tab20b
tab20c


Miscellaneous colormaps
flag
prism
ocean
gist_earth
terrain
gist_stern
gnuplot
gnuplot2
CMRmap
cubehelix
brg
gist_rainbow
rainbow
jet
turbo
nipy_spectral
gist_ncar

matplot 绘图demo

```
fig = plt.figure(figsize=(16, 8),dpi=100)
plt.subplots_adjust(left=0.1, bottom=0.1, right=0.9, top=0.9, wspace=0.4, hspace=0.4)

ax1 = fig.add_subplot(121)
ax1.set_title('bpg ground roll patch')
ax1.imshow(hyper, cmap=mm, vmax=clip, vmin=-clip,aspect=0.05)

ax2 = fig.add_subplot(122)
ax2.set_title('denoise patch')
ax2.imshow(gwden, cmap=mm, vmax=clip, vmin=-clip,aspect=0.05)

plt.show()
```

- 自定义colormap示意

```
# ==================================================================================
#    Copyright (C) 2024 Chengdu University of Technology.
#    Copyright (C) 2024 Zifei Li.
#
#    Filename: seis.py
#    Author: Zifei Li
#    Institute: Chengdu University of Technology
#    Email: 202005050218@stu.cdut.edu.cn
#    Work: 2024/05/20/
#    Function:
#
#    This program is free software: you can redistribute it and/or modify it
#    under the terms of the GNU General Public License as published by the Free
#    Software Foundation, either version 3 of the License, or an later version.
#=================================================================================
import numpy as np
import math
import torch
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors

def seis(input):
    N=40
    L=40
    if input == 1:  #(black-brown)
        u1 = np.concatenate((0.5 * np.ones(N), np.linspace(0.5, 1, 128-N), np.linspace(1, 0, 128-N), np.zeros(N)))
        u2 = np.concatenate((0.25 * np.ones(N), np.linspace(0.25, 1, 128-N), np.linspace(1, 0, 128-N), np.zeros(N)))
        u3 = np.concatenate((np.zeros(N), np.linspace(0, 1, 128-N), np.linspace(1, 0, 128-N), np.zeros(N)))
    elif input == 2: #(black-red)
        u1 = np.concatenate((np.ones(N), np.linspace(1, 1, 128-N), np.linspace(1, 0, 128-N), np.zeros(N)))
        u2 = np.concatenate((np.zeros(N), np.linspace(0, 1, 128-N), np.linspace(1, 0, 128-N), np.zeros(N)))
        u3 = np.concatenate((np.zeros(N), np.linspace(0, 1, 128-N), np.linspace(1, 0, 128-N), np.zeros(N)))
    elif input == 3: #(blue-red)
        u1 = np.concatenate((np.zeros(N), np.linspace(0., 1, 128 - N - L//2), np.ones(L), np.linspace(1, 0.5, 128 - L//2)))
        u2 = np.concatenate((np.zeros(N), np.linspace(0., 1, 128 - N - L//2), np.ones(L), np.linspace(1, 0., 128 - N - L//2), np.zeros(N)))
        u3 = np.concatenate((np.linspace(0.5, 1, 128 - L//2), np.ones(L), np.linspace(1, 0., 128 - N - L//2), np.zeros(N)))

    M = np.column_stack((u1, u2, u3))
    # 创建自定义的colormap
    custom_colormap = mcolors.ListedColormap(M)
    return custom_colormap
```

Chapter4:Useritem

Linux杂七杂八的东西

cuda安装与路径配置:

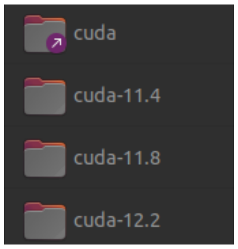
1. .deb安装

- `wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-ubuntu2204.pin`
- `sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-repository-pin-600`
- `wget https://developer.download.nvidia.com/compute/cuda/12.2.1/local_installers/cuda-repo-ubuntu2204-12-2-local_12.2.1-535.86.10-1_amd64.deb`
- `sudo dpkg -i cuda-repo-ubuntu2204-12-2-local_12.2.1-535.86.10-1_amd64.deb`
- `sudo cp /var/cuda-repo-ubuntu2204-12-2-local/cuda-*-keyring.gpg /usr/share/keyrings/`
- `sudo apt-get update`
- `sudo apt-get -y install cuda`
  .deb格式的安装我一直不知道怎么指定安装路径

1. .run安装
   这个格式可以指定安装路径，先更改Toolkit Options (/usr这种非用户目录的都要去掉，我这里全去掉了，另外进入 Change Toolkit Install Path设置cuda安装到自己具有写入权限的路径（提前建好），我这里是“/home/Softwares/...”)
   环境配置如下：

```
# >>> cuda env <<<
export LD_LIBRARY_PATH=/where/you/cuda/install/lib64:/usr/local/cuda/extras/CPUTI/lib64
export CUDA_HOME=/where/you/cuda/install/bin
export PATH=$PATH:$LD_LIBRARY_PATH:$CUDA_HOME
```

一台机子可以装多个版本的cuda，只要把cuda软链接到不同版本的cuda安装主文件夹就行，所以在安装cuda的时候切忌默认安装文件夹cuda，会覆盖多个版本，自己手动给安装文件夹带个后缀哦。



dpkg

- 安装软件
  `dpkg -i <.deb file name>`

示例： `dpkg -i avg71flm_r28-1_i386.deb`

- 安装一个目录下面所有的软件包
  `dpkg -R`

示例： `dpkg -R /usr/local/src`

- 释放软件包，但是不进行配置
  `dpkg –unpack package_file 如果和-R一起使用，参数可以是一个目录`

示例： `dpkg –unpack avg71flm_r28-1_i386.deb`

- 重新配置和释放软件包
  `dpkg –configure package_file`

如果和-a一起使用，将配置所有没有配置的软件包
示例： `dpkg –configure avg71flm_r28-1_i386.deb`

- 删除软件包（保留其配置信息）

`dpkg -r`

示例：`dpkg -r avg71flm`

- 替代软件包的信息
  `dpkg –update-avail <Packages-file`
- 合并软件包信息
  `dpkg –merge-avail <Packages-file`
- 从软件包里面读取软件的信息
  `dpkg -A package_file
- 删除一个包（包括配置信息）
  `dpkg -P`
- 丢失所有的Uninstall的软件包信息
  `dpkg –forget-old-unavail`
- 删除软件包的Avaliable信息
  `dpkg –clear-avail`
- 查找只有部分安装的软件包信息
  `dpkg -C`
- 比较同一个包的不同版本之间的差别
  `dpkg –compare-versions ver-op ver2`
- 显示帮助信息
  `dpkg –help`
- 显示dpkg的Licence
  `dpkg –licence (or) dpkg –license`
- 显示dpkg的版本号
  `dpkg –version`
- 建立一个deb文件
  `dpkg -b direc×y [filename]`
- 显示一个Deb文件的目录
  `dpkg -c filename`
- 显示一个Deb的说明
  `dpkg -I filename [control-file]`
- 搜索Deb包
  `dpkg -l package-name-pattern`
  示例：`dpkg -I vim`
- 显示所有已经安装的Deb包，同时显示版本号以及简短说明
  `dpkg -l`
- 报告指定包的状态信息
  `dpkg -s package-name`
  示例：`dpkg -s ssh`
- 显示一个包安装到系统里面的文件目录信息
  `dpkg -L package-Name`
  示例：`dpkg -L apache2`
- 搜索指定包里面的文件（模糊查询）
  `dpkg -S filename-search-pattern`
- 显示包的具体信息
  `dpkg -p package-name`
  示例：`dpkg -p cacti`
- 指定安装路径
  `dpkg

Nvidia驱动

1. 安装
   官网下载对应型号的显卡驱动

   禁用独立显卡

   ```
   sudo -s
   ./*.run
   ```

2. 动态查看进程

   ```
   watch -n 2 -d nvidia-smi
   ```

进程中断

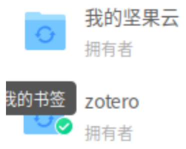
```
kill -9 -PID
```

Zotero/Zotero7文献管理

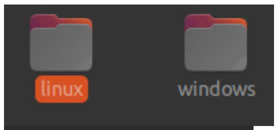
Zotero是一个强大的开源文献管理软件，支持相当多的插件开发，这些插件能很好地帮助用户科研阅读。

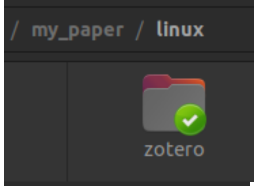
坚果云-zotero同步（稳定方法）

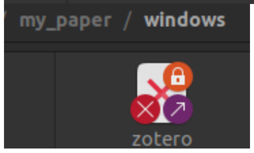
1. 注册坚果云并建立一个同步文件夹命名为zotero

2. 分别在本地建立linux-windows两个文件夹，每个文件夹下包含zotero文件夹，并同步到云端的zotero文件

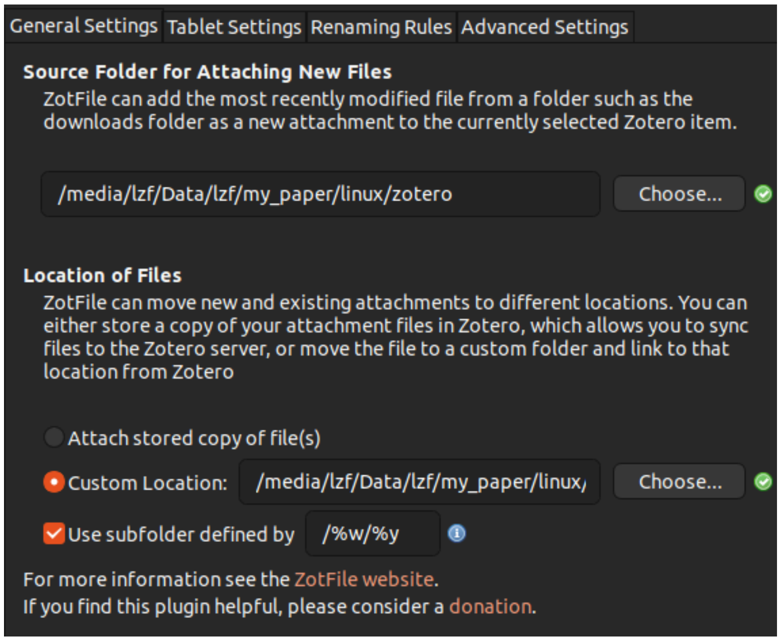
   

   如果不采取上面的形式，部分操作系统在同步文件夹的时候会出现图3所示的情况，为一个链接形式，所以分开系统同步文件夹是不错的选择

3. 安装ZotFile github开源

   工具-ZotFile preference



Source Folder for Attaching New Files
ZotFile can add the most recently modified file from a folder such as the downloads folder as a new attachment to the currently selected Zotero item.

/media/lzf/Data/lzf/my_paper/linux/zotero

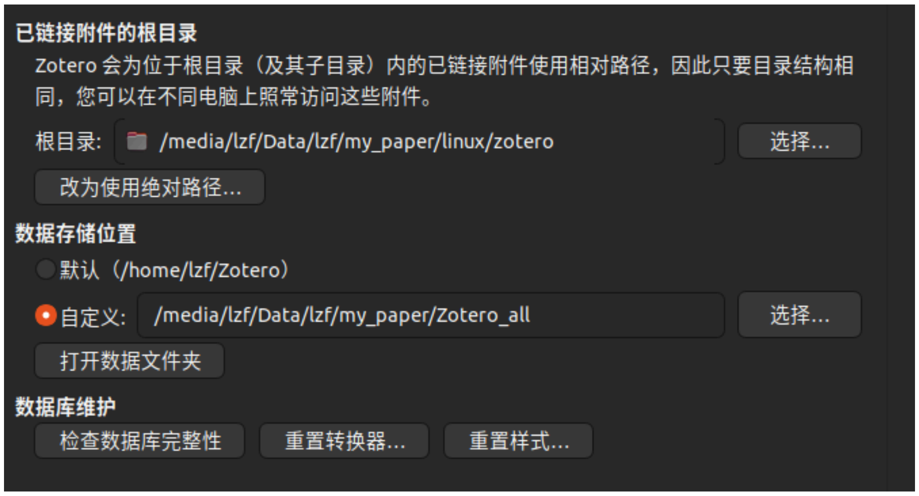
Location of Files
ZotFile can move new and existing attachments to different locations. You can either store a copy of your attachment files in Zotero, which allows you to sync files to the Zotero server, or move the file to a custom folder and link to that location from Zotero

Attach stored copy of file(s)
[✓ Custom Location:] /media/lzf/Data/lzf/my_paper/linux/

[✓ Use subfolder defined by] /%w/%y

For more information see the ZotFile website.
If you find this plugin helpful, please consider a donation.

下面一个文件夹链接到同步文件夹，该文件夹用于存放使用Zotero导出的pdf文件夹，上面一个链接可选可不选，上面一个链接为自动将链接内的pdf打为ZotFile标签。



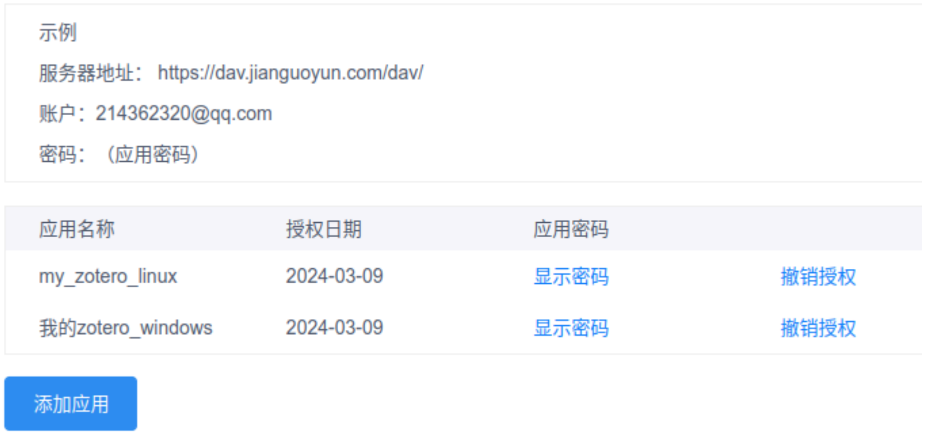
已链接附件的根目录
Zotero 会为位于根目录（及其子目录）内的已链接附件使用相对路径，因此只要目录结构相同，您可以在不同电脑上照常访问这些附件。
根目录: /media/lzf/Data/lzf/my_paper/linux/zotero

数据存储位置
默认（/home/lzf/Zotero）
[✓ 自定义:] /media/lzf/Data/lzf/my_paper/Zotero_all

数据库维护

编辑-首选项-高级-文件和文件夹，分别进行图示配置，上面一个路径存放同步文件夹位置，用于在不同系统下读取相同相对路径格式的文件夹,因为我们共享文件夹库是用ZotFile生成的，默认格式完全相同，这样可以实现在不同操作系统下的文献读取。下面一个位置是Zotero的所有插件保存文件夹，两个系统需要保证文件夹的路径完全一样，当然也可以使用zotero自动同步的功能，插件的大小很小，不影响。

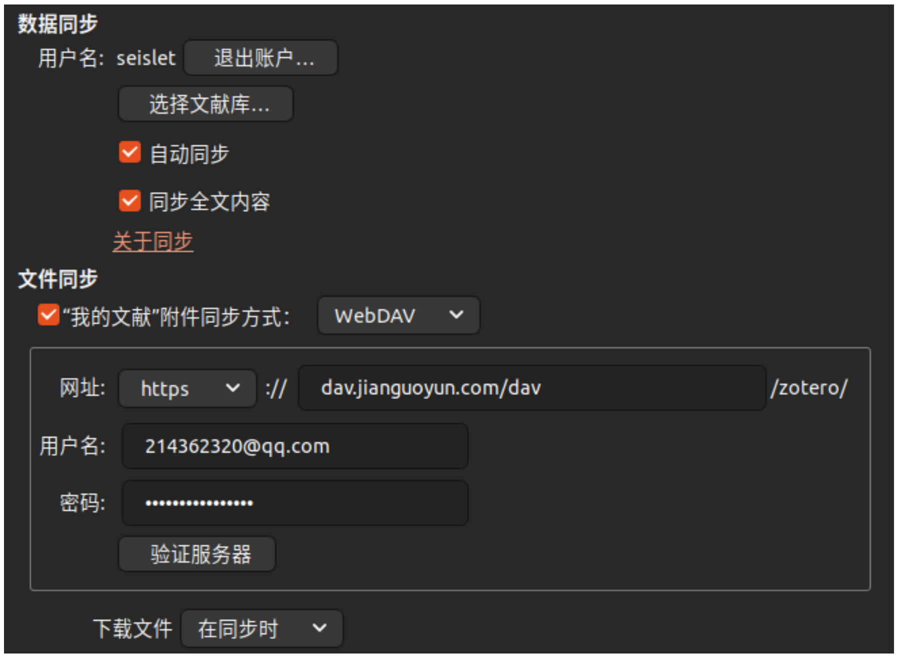
4. 设置坚果云下载同步
该步骤按理论说采取本方法是不需要的，但是为了保险起见仍然在这里记载一下配置方法。坚果云用户-账户信息-安全选项，给你的zotero授权。



示例
服务器地址：https://dav.jianguoyun.com/dav/
账户：214362320@qq.com
密码：（应用密码）

应用名称	授权日期	应用密码	
my_zotero_linux	2024-03-09	显示密码	撤销授权
我的zotero_windows	2024-03-09	显示密码	撤销授权

编辑-首选项-同步，根据坚果云网页所给的密码信息添加授权


数据同步

用户名：seislet

文件同步

✓ "我的文献"附件同步方式：WebDAV

网址：https :// dav.jianguoyun.com/dav /zotero/

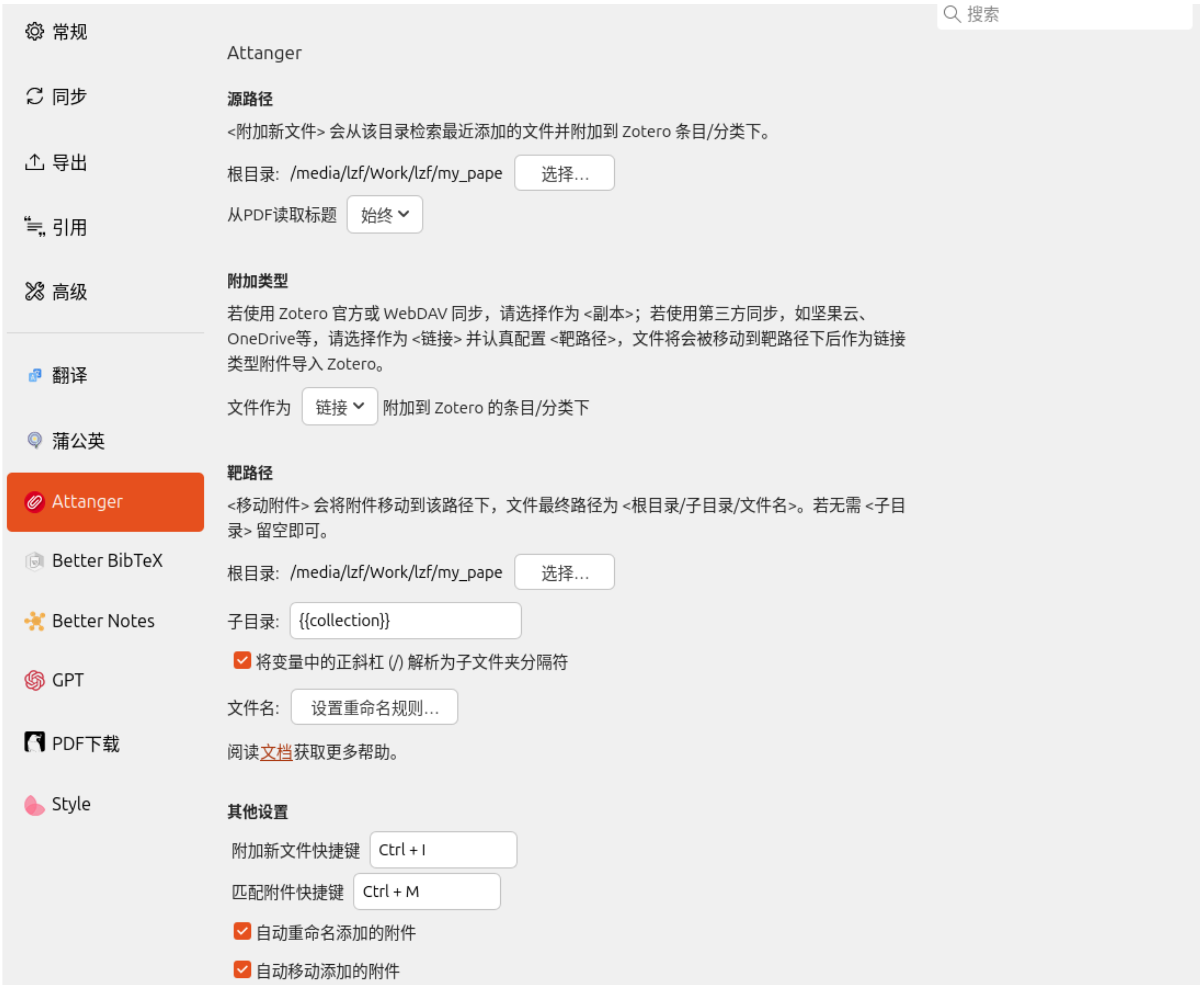
用户名：214362320@qq.com

密码：••••••••••••••••

下载文件 在同步时

bilibili单系统配置教程：
【文献管理软件Zotero详细教程四（如何实现与坚果云的云同步）】https://www.bilibili.com/video/BV1cP411N766/
可配合文案与视频理解。
经一位朋友的分享，有兴趣进一步了解云同步相关知识的同学，可以看看下列文章：
https://www.zhihu.com/question/279410792/answer/1105909839

zotero 7版本的同步和上述相同，只不过zotofile插件变成了Attanger，其余设置方法完全相同



Attanger

源路径

<附加新文件> 会从该目录检索最近添加的文件并附加到 Zotero 条目/分类下。

根目录: /media/lzf/Work/lzf/my_pape 选择…

从PDF读取标题 始终

附加类型

若使用 Zotero 官方或 WebDAV 同步，请选择作为 <副本>；若使用第三方同步，如坚果云、OneDrive等，请选择作为 <链接> 并认真配置 <靶路径>，文件将会被移动到靶路径下后作为链接类型附件导入 Zotero。

文件作为 链接 附加到 Zotero 的条目/分类下

靶路径

<移动附件> 会将附件移动到该路径下，文件最终路径为 <根目录/子目录/文件名>。若无需 <子目录> 留空即可。

根目录: /media/lzf/Work/lzf/my_pape 选择…

子目录: {{collection}}

[✓] 将变量中的正斜杠 (/) 解析为子文件夹分隔符

文件名: 设置重命名规则…

阅读文档获取更多帮助。

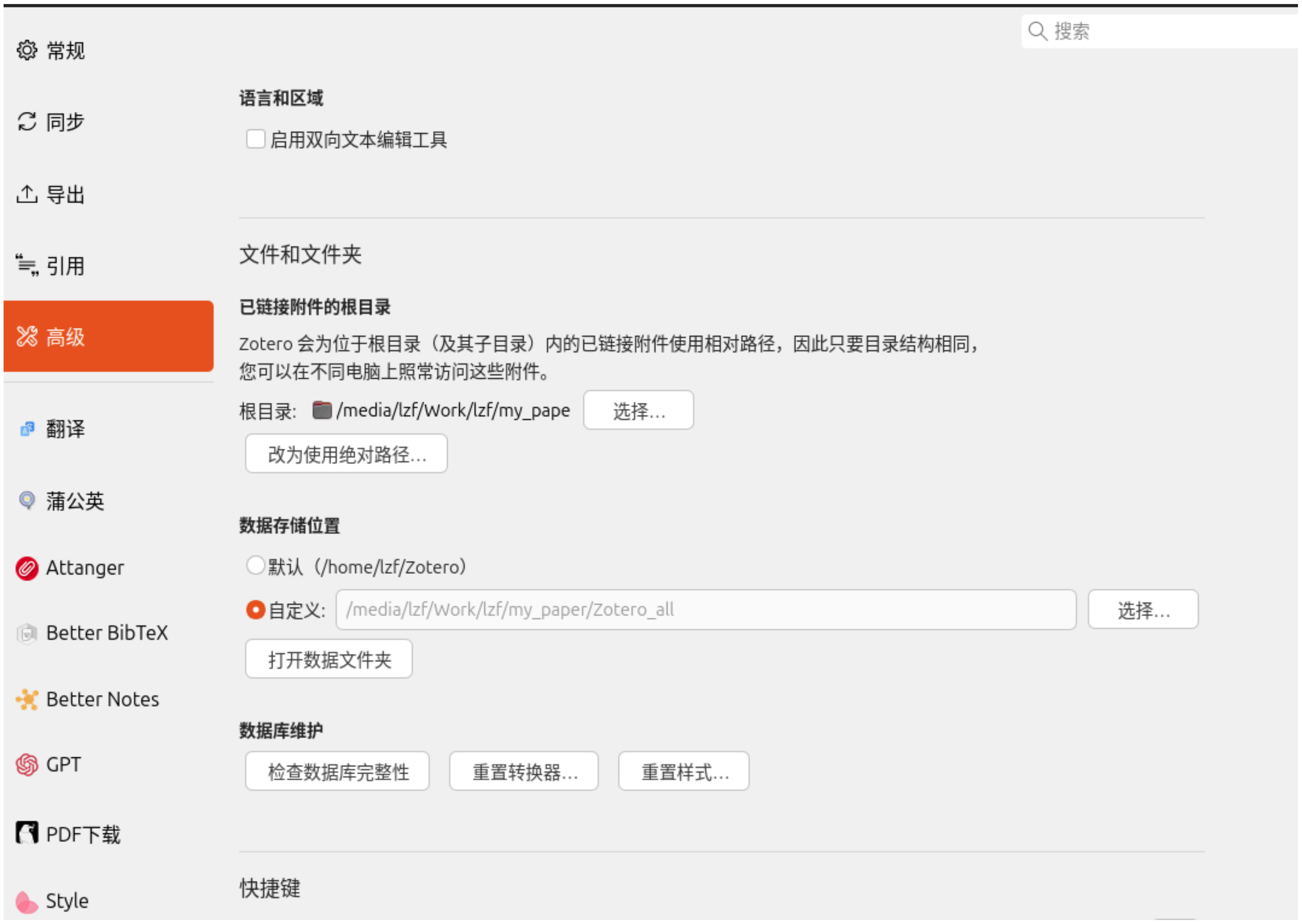
其他设置

附加新文件快捷键 Ctrl + I

匹配附件快捷键 Ctrl + M

[✓] 自动重命名添加的附件

[✓] 自动移动添加的附件



搜索

常规
同步
导出
引用
高级
翻译
蒲公英
Attanger
Better BibTeX
Better Notes
GPT
PDF下载
Style

语言和区域

☐ 启用双向文本编辑工具

文件和文件夹

已链接附件的根目录

Zotero 会为位于根目录（及其子目录）内的已链接附件使用相对路径，因此只要目录结构相同，您可以在不同电脑上照常访问这些附件。

根目录: /media/lzf/Work/lzf/my_pape 选择…

改为使用绝对路径…

数据存储位置

默认（/home/lzf/Zotero） [✓ 自定义: /media/lzf/Work/lzf/my_paper/Zotero_all] 选择…

打开数据文件夹

数据库维护

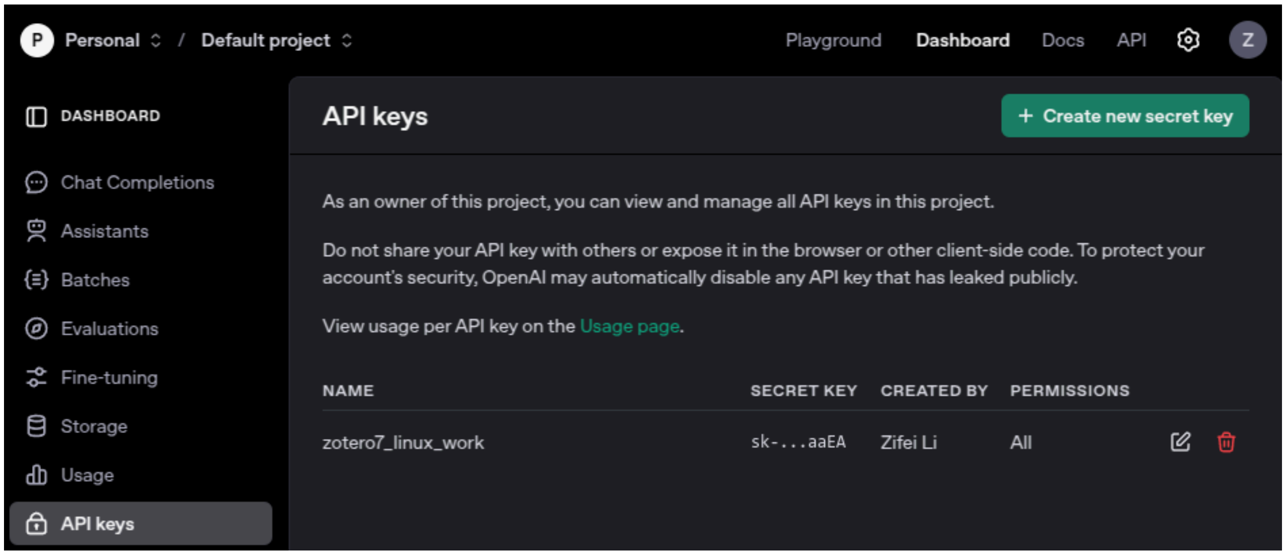
检查数据库完整性 重置转换器… 重置样式…

快捷键

配置AI

配置AI的核心插件为Awesome GPT(https://zotero-chinese.com/user-guide/plugins/zotero-gpt.html)
具体操作如下（以deepseek+通义千问为例）

1. 配置api接口，在所需配置的大模型网站找到ai接口，如openai网站的(https://platform.openai.com/api-keys):



P Personal / Default project   Playground Dashboard Docs API Z

DASHBOARD
Chat Completions
Assistants
Batches
Evaluations
Fine-tuning
Storage
Usage
API keys

API keys   + Create new secret key

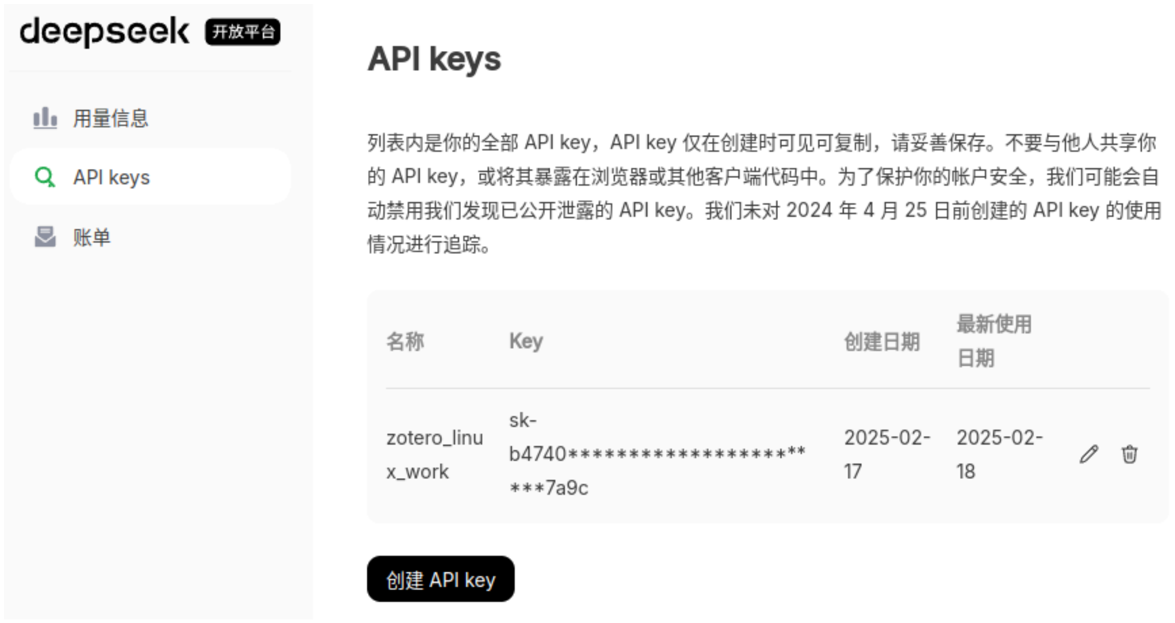
As an owner of this project, you can view and manage all API keys in this project.

Do not share your API key with others or expose it in the browser or other client-side code. To protect your account's security, OpenAI may automatically disable any API key that has leaked publicly.

View usage per API key on the Usage page.

NAME	SECRET KEY	CREATED BY	PERMISSIONS
zotero7_linux_work	sk-...aaEA	Zifei Li	All

deepseek网站的(https://platform.deepseek.com/api_keys):



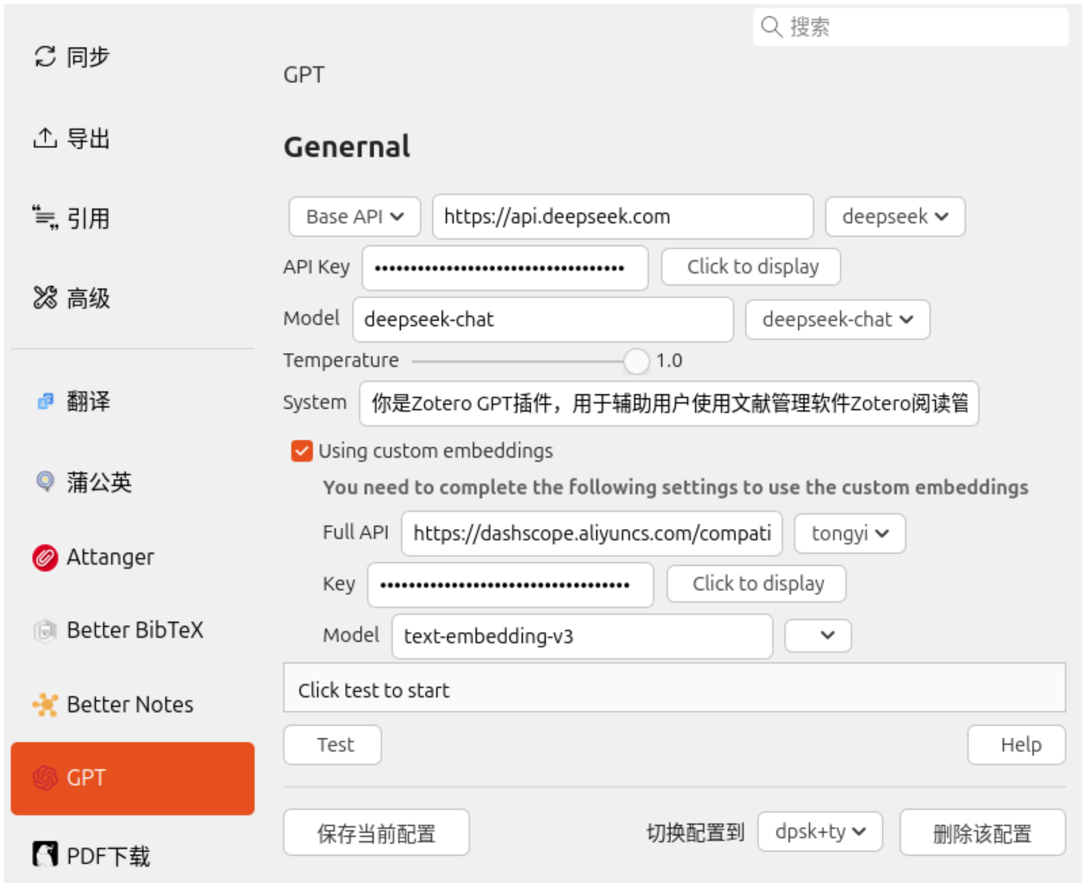
API keys

列表内是你的全部 API key，API key 仅在创建时可见可复制，请妥善保存。不要与他人共享你的 API key，或将其暴露在浏览器或其他客户端代码中。为了保护你的帐户安全，我们可能会自动禁用我们发现已公开泄露的 API key。我们未对 2024 年 4 月 25 日前创建的 API key 的使用情况进行追踪。

名称	Key	创建日期	最新使用日期
zotero_linux_work	sk-b4740*********************7a9c	2025-02-17	2025-02-18

在网站上创建一个api key，复制密码字符

2. 在Zotero-编辑-设置中找到GPT设置，按照图示配置



GPT

Genernal

Base API: https://api.deepseek.com — deepseek
API Key: •••••••••••••••••••••••••••••••••• Click to display
Model: deepseek-chat — deepseek-chat
Temperature: 1.0
System: 你是Zotero GPT插件，用于辅助用户使用文献管理软件Zotero阅读管

☑ Using custom embeddings

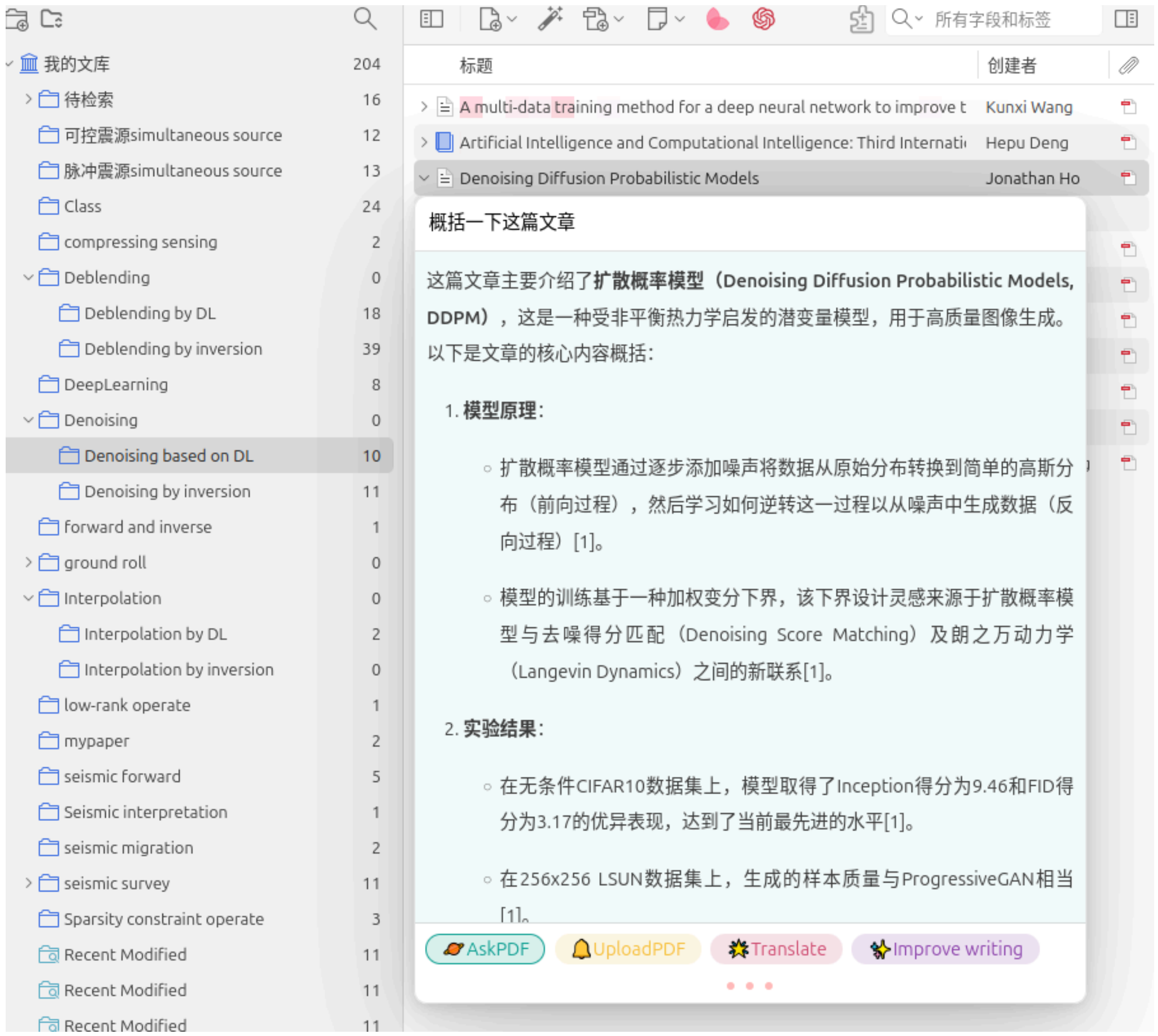
You need to complete the following settings to use the custom embeddings

Full API: https://dashscope.aliyuncs.com/compati — tongyi
Key: •••••••••••••••••••••••••••••••••• Click to display
Model: text-embedding-v3

Click test to start

Base API中设置大模型链接，API key粘贴进去，然后选择大模型，其中Tem控制只是密度，embedding选项为是否设置通用的向量支持模型，由于dpsk不支持向量模型所以另选通义千问来设置，设置方法和上述步骤一样。如果选择支持向量模型的大模型那这块就不用设置。

配置好了之后就可进行如下的ai辅助:



我的文库 204
待检索 16
可控震源simultaneous source 12
脉冲震源simultaneous source 13
Class 24
compressing sensing 2
Deblending 0
Deblending by DL 18
Deblending by inversion 39
DeepLearning 8
Denoising 0
Denoising based on DL 10
Denoising by inversion 11
forward and inverse 1
ground roll 0
Interpolation 0
Interpolation by DL 2
Interpolation by inversion 0
low-rank operate 1
mypaper 2
seismic forward 5
Seismic interpretation 1
seismic migration 2
seismic survey 11
Sparsity constraint operate 3
Recent Modified 11
Recent Modified 11
Recent Modified 11

标题	创建者
A multi-data training method for a deep neural network to improve t	Kunxi Wang
Artificial Intelligence and Computational Intelligence: Third Internati	Hepu Deng
Denoising Diffusion Probabilistic Models	Jonathan Ho

概括一下这篇文章

这篇文章主要介绍了**扩散概率模型（Denoising Diffusion Probabilistic Models, DDPM）**，这是一种受非平衡热力学启发的潜变量模型，用于高质量图像生成。以下是文章的核心内容概括：

1. **模型原理**：
   - 扩散概率模型通过逐步添加噪声将数据从原始分布转换到简单的高斯分布（前向过程），然后学习如何逆转这一过程以从噪声中生成数据（反向过程）[1]。
   - 模型的训练基于一种加权变分下界，该下界设计灵感来源于扩散概率模型与去噪得分匹配（Denoising Score Matching）及朗之万动力学（Langevin Dynamics）之间的新联系[1]。
2. **实验结果**：
   - 在无条件CIFAR10数据集上，模型取得了Inception得分为9.46和FID得分为3.17的优异表现，达到了当前最先进的水平[1]。
   - 在256x256 LSUN数据集上，生成的样本质量与ProgressiveGAN相当[1]。

pandoc

```
pandoc.exe test.md -f markdown -t html -s -o test.html
```

查看ip

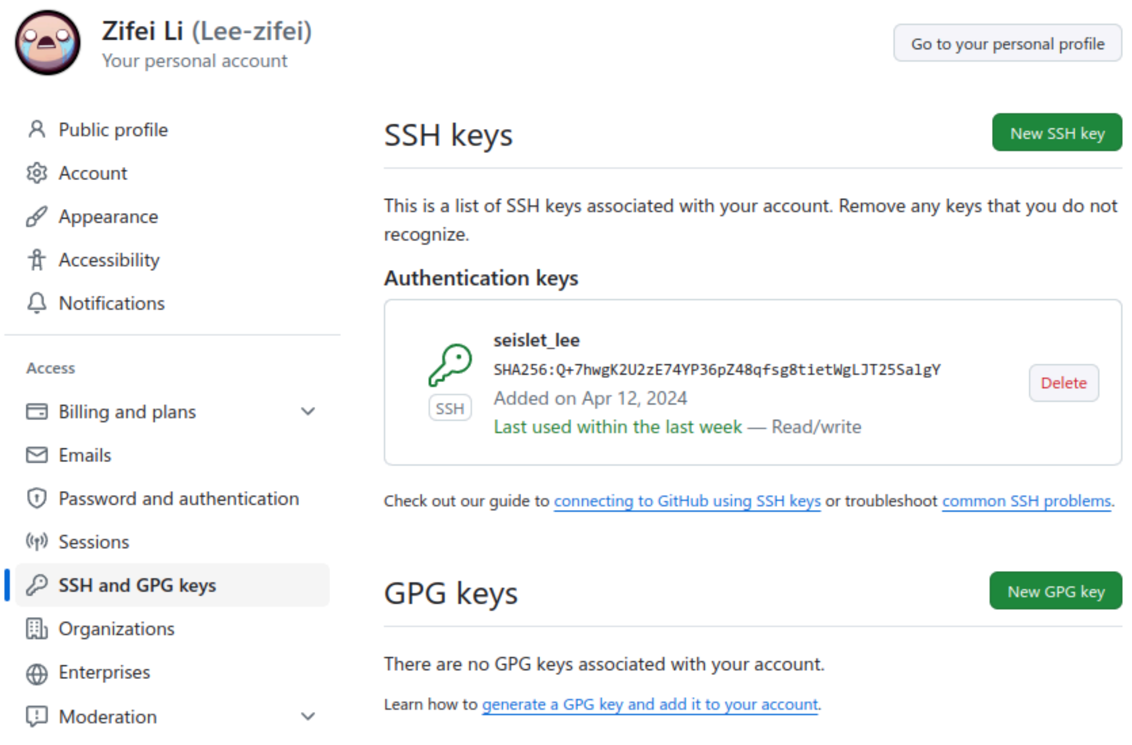
ifconfig

VPN

github本地上传

前置条件

创建ssh链接并拷贝到github ssh上
生成密钥对： `ssh-keygen`
查看密钥对： `cat ~/.ssh/id_rsa.pub` 默认位置在/home/lzf/.ssh下
github授权：



SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

seislet_lee
SHA256:Q+7hwgK2U2zE74YP36pZ48qfsg8tietWgLJT25SalgY
Added on Apr 12, 2024
Last used within the last week — Read/write

Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

本地上传流程

1. 在本地需要上传的文件夹建立git仓库并且初始化
   `git init`
   初始化之后，终端会显示git连接命令： `git:main x [19:46:22] C:number`
2. 添加文件
   `git add 'files'`
   全部添加
   `git add .`
3. 提交改变到缓存
   `git commit -m 'what are you doing'`
4. 本地git仓库关联到github仓库
   `git remote add origin git@github.com:Lee-zifei/zifei.git`

- 如果仓库已经存在链接，但是又是第一次上传，删除链接命令如下： `git remote remove origin`

5. 上传
   `git push -u origin main (--force)`