**Manjaro安装及配置**

**安装虚拟增强**

cd /mnt

sudo mkdir cdrom

设备-虚拟增强

sudo mount /dev/cdrom cdrom

**1、配置国内镜像源**

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

### 中国

**电信**

* **<http://mirror.bit.edu.cn/archlinux/>** - *北京理工大学*
* **<http://mirrors.aliyun.com/archlinux/>** - *阿里巴巴*
* **<http://mirrors.163.com/archlinux/>** - *网易*

**联通**

* **<http://mirrors.sohu.com/archlinux/>**
* **<http://mirrors.yun-idc.com/archlinux/>**

**教育网**

* **<http://mirror.hust.edu.cn/archlinux/>** - *华中科技大学*
* **<http://ftp.sjtu.edu.cn/archlinux/>** - *上海交通大学*
* **<http://mirrors.4.tuna.tsinghua.edu.cn/archlinux/>** *(ipv4 only)*
* **<http://mirrors.6.tuna.tsinghua.edu.cn/archlinux/>** *(ipv6 only)*
* **<http://mirror.lzu.edu.cn/archlinux/>** - *兰州大学*
* **<https://mirrors.nju.edu.cn/archlinux/>** - *南京大学*
* **<https://mirrors.dgut.edu.cn/archlinux/>** - *东莞理工学院*

**设置 archlinuxcn 源**

sudo nano /etc/pacman.conf

## 添加以下内容,此源很快，放在文件最上面

[archlinuxcn]

SigLevel = Optional TrustedOnly

Server=https://mirrors.ustc.edu.cn/archlinuxcn/$arch

#Server = <http://repo.archlinuxcn.org/$arch>

<https://www.archlinux.org/mirrorlist/##>生成源的网站

## China

#Server = http://mirrors.163.com/archlinux/$repo/os/$arch

#Server = http://mirrors.cqu.edu.cn/archlinux/$repo/os/$arch

#Server = https://mirrors.cqu.edu.cn/archlinux/$repo/os/$arch

#Server = http://mirror.lzu.edu.cn/archlinux/$repo/os/$arch

#Server = http://mirrors.neusoft.edu.cn/archlinux/$repo/os/$arch

#Server = https://mirrors.neusoft.edu.cn/archlinux/$repo/os/$arch

#Server = https://mirrors.sjtug.sjtu.edu.cn/archlinux/$repo/os/$arch

#Server = http://mirrors.tuna.tsinghua.edu.cn/archlinux/$repo/os/$arch

#Server = https://mirrors.tuna.tsinghua.edu.cn/archlinux/$repo/os/$arch

#Server = http://mirrors.ustc.edu.cn/archlinux/$repo/os/$arch

#Server = https://mirrors.ustc.edu.cn/archlinux/$repo/os/$arch

#Server = https://mirrors.xjtu.edu.cn/archlinux/$repo/os/$arch

#Server = http://mirrors.zju.edu.cn/archlinux/$repo/os/$arch

**完成后执行下面的命令使配置生效**

## 更新源列表

sudo pacman-mirrors -g

## 更新pacman数据库并全面更新系统

sudo pacman -Syyu

##更名

Alias upgrade= “sudo pacman -Syu”

##安装**archlinuxcn-keyring**，安装完特别快，速度月5-6m/s，更新后必须安装密钥

sudo pacman -S archlinuxcn-keyring

配置pacman

1．查看旧版和新版的有效安装包，请取消/etc/pacman.conf中"VerbosePkgLists"的注释

2.Pacman 具有颜色选项，取消 "Color" 行的注释即可

3.用户不希望升级某个软件包，可以加入内容如下：IgnorePkg = 软件包名;IgnoreGroup = 包组。多软件包可以用空格隔开，也可是用 glob 模式。如果只打算忽略一次升级，可以使用 --ignore 选项。忽略了的软件包可通过 pacman -S 升级。

4. 如果你有多个配置文件（比如，主配置和测试testing仓库生效的配置文件），需要共享一些设置：Include = /path/to/common/settings，此文件中是两个配置文件共享的相同配置。

**pacman软件[包管理器](https://en.wikipedia.org/wiki/Package_management_system" \o "wikipedia:Package management system)**是 Arch Linux 的一大亮点。它将一个简单的二进制包格式和易用的构建系统结合了起来(参见**[makepkg](https://wiki.archlinux.org/index.php/Makepkg" \o "Makepkg)**和**[ABS](https://wiki.archlinux.org/index.php/ABS" \o "ABS)**)。不管软件包是来自官方的 Arch 库还是用户自己创建，*pacman* 都能方便地管理。

*pacman* 通过和主服务器同步软件包列表来进行系统更新。这种服务器/客户端模式可以使用一条命令就下载或安装软件包，同时安装必需的依赖包。

*pacman* 用 C 语言编写，使用**[tar](https://en.wikipedia.org/wiki/tar_(computing)" \o "w:tar (computing))**打包格式。

**注意:**软件包通常有很多**[可选依赖](https://wiki.archlinux.org/index.php/PKGBUILD" \l "optdepends" \o "PKGBUILD)**， 它们为软件提供额外功能， 并不强制要求安装它们。 安装软件时, *pacman* 将会输出它的可选依赖, 但是这个输出不会在 pacman.log中；当你想浏览已安装软件的可选依赖时可以使用pacman -Si ，得到关于可选依赖的简短描述。

**警告:**在Arch下安装软件包时，未**[更新](https://wiki.archlinux.org/index.php/Pacman_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)" \l "%E5%8D%87%E7%BA%A7%E8%BD%AF%E4%BB%B6%E5%8C%85)**系统前，**不要**更新软件包数据库（例如，可能出现某软件包不再出现在官方库）。操作时，应使用pacman -Syu *package\_name*, 而不要使用（pacman -Sy *package\_name*），否则可能会有依赖问题

**注意: 如果列表中的包已经安装在系统中，它会被重新安装，即使它已经是最新的。可以用 --needed 选项覆盖这种行为。**

注意: pacman 不会删除软件自己创建的文件(例如主目录中的 .dot 文件不会被删除。

pacman 删除某些程序时会备份重要配置文件，在其后面加上\*.pacsave扩展名。-n 选项可以避免备份这些文件：

警告: \* 建议所有用户都 [经常性的更新系统](https://wiki.archlinux.org/index.php/System_maintenance" \l "Upgrading_the_system" \o "System maintenance)。Arch 只支持系统完整升级。

一个 pacman 命令就可以升级整个系统。花费的时间取决于系统有多老。

这个命令会同步非本地(local)软件仓库并升级系统的软件包。

pacman数据库通常位于 /var/lib/pacman/sync. 对于每一个在/etc/pacman.conf中指定的软件仓库， 这里都有一个一致的数据库。数据库文件夹里每个tar.gz文件都包含着一个仓库的软件包信息。例如[which](https://www.archlinux.org/packages/?name=which) 包:

% tree which-2.20-6

which-2.20-6

|-- depends

-- desc

这个 depends 项列出了该软件的依赖包， 而desc有该包的介绍，例如文件大小和MD5值 。

清理软件包缓存

pacman 将下载的软件包保存在 /var/cache/pacman/pkg/ 并且不会自动移除旧的和未安装版本的软件包，因此需要手动清理，以免该文件夹过于庞大。

使用内建选项即可清除未安装软件包的缓存：

# pacman -Sc

警告:仅在确定当前安装的软件包足够稳定且不需要[降级](https://wiki.archlinux.org/index.php/Downgrading_packages_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)" \o "Downgrading packages (简体中文))时才执行清理。pacman -Sc仅会保留软件包的当前有效版本，旧版本的软件包被清理后，只能从其他地方如 [Arch Linux Archive (简体中文)](https://wiki.archlinux.org/index.php/Arch_Linux_Archive_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)" \o "Arch Linux Archive (简体中文))中获取了。

pacman -Scc 可以清理所有缓存，但这样 pacman 在重装软件包时就只能重新下载了。除非空间不足，否则不应这么做。

由于以上种种局限，建议使用专门的脚本去处理清理哪些、清理多少缓存：

[pacman-contrib](https://www.archlinux.org/packages/?name=pacman-contrib) 提供的 paccache 命令默认会删除近3个版本前的软件包

# paccache -r

Tip: 可以使用 [pacman hooks](https://wiki.archlinux.org/index.php/Pacman_hooks" \o "Pacman hooks) 自动执行清理，[这里](https://bbs.archlinux.org/viewtopic.php?pid=1694743" \l "p1694743)是参考示例。

也可以自己设置保留最近几个版本：

# paccache -rk 1

清理所有未安装包的缓存文件，再此运行paccache:

# paccache -ruk0

更多功能参见paccache -h。

paccache，还可以使用 [Arch User Repository](https://wiki.archlinux.org/index.php/Arch_User_Repository_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)" \o "Arch User Repository (简体中文)) 中的 [pkgcacheclean](https://aur.archlinux.org/packages/pkgcacheclean/)AUR：

# pkgcacheclean

以及[pacleaner](https://aur.archlinux.org/packages/pacleaner/)AUR，这两个是未来的替代工具.

Hooks 钩子

pacman可以在更新安装前后时，运行/usr/share/libalpm/hooks/文件夹下的hooks，更多的hooks文件夹HooDir可以在pacman.conf中设置，默认/etc/pacman.d/hooks。Hook文件必须以.hook结尾。

安装一个本地包(不从源里下载）：

# pacman -U /path/to/package/package\_name-version.pkg.tar.xz

提示： 可以设置一个 crontab 或者 systemd timer 来定期同步文件信息数据库。

pacman 的配置文件位于/etc/pacman.conf。 [man pacman.conf](https://www.archlinux.org/pacman/pacman.conf.5.html) 可以查看配置文件的进一步信息。

软件仓库

除了特殊的[options](https://wiki.archlinux.org/index.php/Pacman_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)" \l "General_options)[[broken link](https://wiki.archlinux.org/index.php/ArchWiki:Requests" \l "Broken_section_links" \o "ArchWiki:Requests): invalid section]section, 每个pacman.conf中的section都定义了一个使用的软件包仓库，仓库是多个软件包的逻辑上的集合，他们物理上存储在一个或多个服务器：这也是为什么每一个服务器都叫做这个仓库的镜像。

仓库区分为[[官方]与[Unofficial user repositories|非官方](https://wiki.archlinux.org/index.php/Official_repositories" \o "Official repositories)两类。配置文件中仓库的顺序十分重要；当几个仓库出现同名安装包，不管版本号是否相同，pacman将使用配置文件中排前的仓库。[upgrade](https://wiki.archlinux.org/index.php/Pacman_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)" \l "Upgrading_packages)[[broken link](https://wiki.archlinux.org/index.php/ArchWiki:Requests" \l "Broken_section_links" \o "ArchWiki:Requests): invalid section]升级整个系统，来让新添加的仓库生效。

每个仓库设置都可以直接指定镜像列表或者Include引用其他的文件：例如，官方镜像引用了/etc/pacman.d/mirrorlist/。具体查看[Mirrors](https://wiki.archlinux.org/index.php/Mirrors" \o "Mirrors)。

软件包的安全性

pacman 4 支持软件包签名。语句 SigLevel = Required DatabaseOptional 将启用全局签名验证，但会被每个软件仓库的 SigLevel 行所覆盖。详情参见 [pacman-key](https://wiki.archlinux.org/index.php/Pacman-key_(%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)" \o "Pacman-key (简体中文)).

**基本中文支持**

要正确显示中文，必需设置正确的locale并安装合适的中文字体。

**locale设置**

安装中文locale

Linux中通过locale来设置程序运行的不同环境。常用的中文locale有（最直观的分别是可显示字的数量）：

zh\_CN.GB2312

zh\_CN.GBK

zh\_CN.GB18030

zh\_CN.UTF-8

zh\_TW.BIG-5

zh\_TW.UTF-8

推荐使用UTF-8的locale。对于glibc（>=2.3.6），需要修改/etc/locale.gen文件来设定系统中可以使用的locale（取消对应项前的注释符号「#」即可）：

en\_US.UTF-8 UTF-8

zh\_CN.UTF-8 UTF-8

然后执行locale-gen命令，便可以在系统中使用这些locale。可以通过locale命令来查看当前使用的locale：亦可通过locale -a命令来查看目前可以使用的locale；

启用中文locale

Arch Linux中，通过/etc/locale.conf文件设置全局有效的locale：

LANG=en\_US.UTF-8

**警告: 不推荐在此设置中文locale，会导致tty乱码，在tty下亦可显示和输入中文，但需要安装cce、zhcon或[fbterm](https://github.com/felixonmars/aur3-mirror/tree/master/fbterm)；其实可以，一劳永逸。**

提示： 可为内核打中文补丁；

对于特定用户，还可以在~/.bashrc、~/.xinitrc或~/.xprofile中设置自己的用户环境。不同之处在于：

.bashrc: 每次终端登录时读取并运用里面的设置。

.xinitrc: 每次startx启动X界面时读取并运用里面的设置

.xprofile: 每次使用gdm(GNOME显示管理器)等图形登录时读取并运用里面的设置

**单独在图形界面启用中文locale**

不推荐/etc/locale.conf使用全局中文locale，会导致tty乱码。

如前面所说，可以在~/.xinitrc或~/.xprofile单独设置中文locale。添加如下内容到上述文件最前端注释之后（如果不确定使用哪个文件，可以都添加）：

export LANG=zh\_CN.UTF-8

export LANGUAGE=zh\_CN:en\_US

export LC\_CTYPE=en\_US.UTF-8

注意: 若欲将此三行放至~/.xinitrc中，请注意将其放在exec \_example\_WM\_or\_DE\_行之前；此为常见错误；

注意: 该方法适用于slim或者无登陆管理器的用户，GDM和KDM用户可以在Gnome或KDE设置中选择语言。

**图形界面用户设置全面的中文**

添加如下内容到文件~/.xprofile文件中

export LC\_ALL="zh\_CN.UTF-8"

注意: 图形界面用户使用

安装完中文环境、输入法后，不能用中文或显示为框框，需要安装中文字体

### 中文字体

#### 安装字体

除了设置好locale，还需要安装中文字体。

常用的免费（GPL或兼容版权）中文字体有：

* **[wqy-microhei](https://www.archlinux.org/packages/?name=wqy-microhei)**
* **[wqy-microhei-lite](https://www.archlinux.org/packages/?name=wqy-microhei-lite)**
* **[wqy-bitmapfont](https://www.archlinux.org/packages/?name=wqy-bitmapfont)**
* **[wqy-zenhei](https://www.archlinux.org/packages/?name=wqy-zenhei)**
* **[ttf-arphic-ukai](https://www.archlinux.org/packages/?name=ttf-arphic-ukai)**
* **[ttf-arphic-uming](https://www.archlinux.org/packages/?name=ttf-arphic-uming)**
* **[adobe-source-han-sans-cn-fonts](https://www.archlinux.org/packages/?name=adobe-source-han-sans-cn-fonts)**
* **[adobe-source-han-serif-cn-fonts](https://www.archlinux.org/packages/?name=adobe-source-han-serif-cn-fonts)**
* **[noto-fonts-cjk](https://www.archlinux.org/packages/?name=noto-fonts-cjk)**

系统字体将默认安装到/usr/share/fonts。如果没有root权限或只打算自己使用某些字体，可以直接复制这些字体到~/.fonts目录（或其子目录）下面，并把该路径加入/etc/fonts/local.conf中

**安装中文字体**

sudo pacman -S --noconfirm powerline-fonts awesome-terminal-fonts wqy-microhei wqy-zenhei ttf-fira-code wqy-microhei-lite wqy-bitmapfont ttf-arphic-ukai ttf-arphic-uming adobe-source-han-sans-cn-fonts adobe-source-han-serif-cn-fonts noto-fonts-cjk && fc-cache -fv

##查看已经安装的字体:  
 fc-list|cut -d: -f2  
##强制刷新字体缓存:  
 fc-cache -fv

**输入法**

**英语**

**Rime中州中州韵**

**搜狗**

**Rime**

**Manjaro使用fcitx-rime挂截小鹤双拼**

**1、安装输入法框架和输入法**

sudo pacman -S fcitx-im #安装fcitx输入法框架，选择全部安装

sudo pacman -S fcitx-configtool #输入法配置文件

sudo pacman -S fcitx-rime #安装rime

添加输入法配置文件 (sudo vim ~/.xprofile)

export GTK\_IM\_MODULE=fcitx

export QT\_IM\_MODULE=fcitx

export XMODIFIERS="@im=fcitx"

重启后输入法生效。

如果不生效，直接复制备份好的配置文件

**2**、进入 <http://flypy.ys168.com/>，下载挂接文件



解压出来是一个rime-date的文件夹

把这个文件夹直接解压到自己的用户目录小的 .config/fcitx下，也就是/home/liming/.config/fcitx

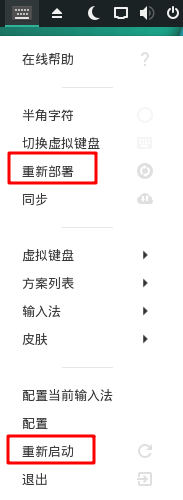
**3**、打开输入法设置，添加中州韵输入法，只保留一个键盘 - 英语(美国)，其他的全部删除，并按照要求把键盘 - 英语(美国)放在最上面



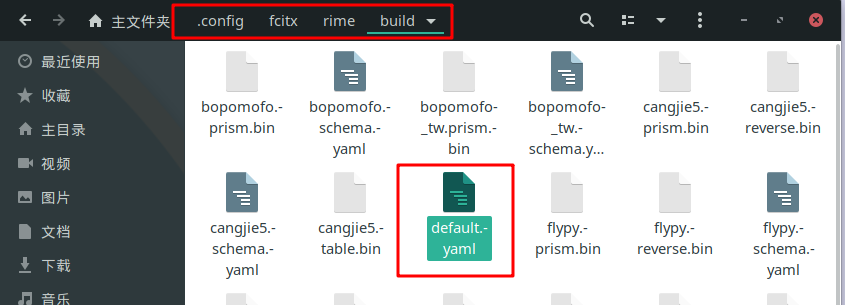
用自带的连打语句流输入法,有备份

**4、**把刚才的rime-date文件夹在原位置复制一份，并重命名为rime

**5、**点击输入法选择器，点击重新启动，再点击重新部署，等待中州韵重新部署完毕。



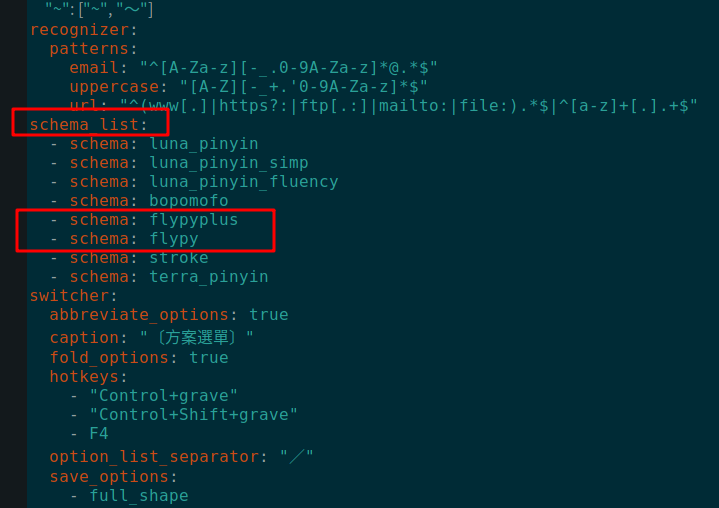
6、进入刚刚重命名的**rime/build文件夹，找到default.yaml**文件，打开



7、找到schema list这一行，在下面按照同样的格式添加两行

flypy

flypyplus



8、重新部署后（重启后）就可以生效了。随便找个输入框，输入一个字符后，按F4或Ctrl+` 就可以选择输入方案了。

如果还不能用，就到输入法配置选择器里面，删除中州韵，再重新添加一遍。

用户入门<https://github.com/rime/home/wiki/UserGuide>

配置

### 语句流风格的输入方案里，以空格键确认后，字词并不立即上屏，而是按下回车键（Return）时整个句子上屏。

### 删除误上屏的错词

不慎上屏了错误的词组，再打同样的编码时，那错词出现在候选栏，令有洁癖的同学十分不爽。这时候可以：

先把选字光标（用上、下键）移到要删除的用户词组上，再按下Shift+Delete或Control+Delete。

只能够从用户词典中删除词组。用于码表中原有的词组时，只会取消其调频效果。

### 备份及合并词典快照

Rime输入法在使用中会在一定时间自动将用户词典备份为快照文件\*.userdb.txt。也可使用管理工具，备份指定的用户词典到快照文件。

备份到快照中的打字习惯，可恢复到新建立的或其他系统上的用户词典。

执行合并词典快照操作，只须选定要合并的.userdb.txt文件，因为快照中已记录了所属词典的名称。

你不必担心来自快照的词条已存在于用户词典的情况；词频更新为二者的较大值，其他参数亦会按照合理的算法叠加。

### 导出及导入文本码表

将用户词典导出为文本码表，便于直接阅读，或批量添加词条到\*.dict.yaml。

导出的文本格式与Rime 静态码表的格式相同：以制表符（Tab）分隔的三列，分别是文字、编码、使用频次。其中，编码是码表中定义的完全形式，多个音节间以空格。

如此导出的文本码表，可反向导入到指定的用户词典。但是，文本码表所含信息不如快照全面，因此为了不损失输入效果，请尽量使用合入快照的方式转移用户资料。

由于文本码表用户可以编辑，导入时无法检查码表内容是否与选定的词典相匹配，请**当心**莫要导入到错误的词典中去了。

共享资料夹

【中州韵】 /usr/share/rime-data/

用户资料夹

【中州韵】 ~/.config/fcitx/rime/

**共享资料夹**包含预设输入方案的源文件。这些文件属于Rime 所发行软件的一部份，在访问权限控制较严格的系统上对用户是只读的，因此谢绝软件版本更新以外的任何修改—— 一旦用户修改这里的文件，很可能影响后续的软件升级或在升级时丢失数据。

在「部署Rime」操作时，将用到这里的输入方案源文件、并结合用户定制的内容来编译预设输入方案。

**用户资料夹**则包含为用户准备的内容，如：

〔全局设定〕 default.yaml

〔发行版设定〕 weasel.yaml

〔预设输入方案副本〕 <方案標識>.schema.yaml

※〔安装信息〕 installation.yaml

※〔用户状态信息〕 user.yaml

编译输入方案所产出的二进制文件：

〔Rime 棱镜〕 <方案標識>.prism.bin

〔Rime 固态词典〕 <詞典名>.table.bin

〔Rime 反查词典〕 <詞典名>.reverse.bin

记录用户写作习惯的文件：

※〔用户词典〕 <詞典名>.userdb.kct

※〔用户词典快照〕<詞典名>.userdb.txt、<詞典名>.userdb.kct.snapshot见于同步文件夹

**以及用户自己设定的：**

**※〔用户对全局设定的定制信息〕 default.custom.yaml**

**※〔用户对预设输入方案的定制信息〕 <方案標識>.custom.yaml**

**※〔用户自制输入方案〕及配套的词典源文件**

注：以上标有※ 号的文件，包含用户资料，您在清理文件时要注意备份！

**详解输入方案**

方案定义

一套输入方案，通常包含「方案定义」和「词典」文件。

方案定义，命名为<方案標識>.schema.yaml，是一份包含输入方案配置信息的YAML文档。

文档中需要有这样一组方案描述：

# 以下代碼片段節選自 luna\_pinyin.schema.yaml

schema:

schema\_id: luna\_pinyin

name: 朙月拼音

version: "0.9"

author:

- 佛振 <chen.sst@gmail.com>

description: |

Rime 預設的拼音輸入方案。

首先来为方案命名。schema/name字段是显示在〔方案选单〕中的名称。

然后——重点是——要定一个在整个Rime输入法中唯一的「方案标识」，即schema/schema\_id字段的内容。方案标识由小写字母、数字、下划线构成。仅于输入法内部使用，且会构成方案定义文件名的一部分，因此为了兼容不同的文件系统，不要用大写字母、汉字、空格和其他符号做方案标识。一例：这款名为【朙月拼音】的输入方案，方案标识为「luna\_pinyin」。

方案如做升级，通过版本号（schema/version）来区分相互兼容的新旧版本。

版本号——以「.」分隔的整数（或文字）构成的字符串。

如下都是版本号常见的形式：

"1" # 最好加引號表明是字符串！

"1.0" # 最好加引號表明是字符串！

"0.9.8"

"0.9.8.custom.86427531" # 這種形式是經過用戶自定義的版本；自動生成

然而，若对方案的升级会导致原有的用户输入习惯无法在新的方案中继续使用，则需要换个新的方案标识。

比如【仓颉五代】之于【仓颉三代】、【五笔98】之于【五笔86】，其实已是互不兼容的输入法。

schema/author ——列出作者和主要贡献者，格式为文字列表：

schema:

author:

- 作者甲 <alpha@rime.org>

- 作者乙 <beta@rime.org>

- 作者丙

schema/description ——对方案作简要介绍的多行文字。

以上schema/schema\_id、schema/version字段用于在程序中识别输入方案，而schema/name、schema/author、schema/description则主要是展示给用户的信息。

除方案描述外，方案定义文件中还包含各种功能设定，控制着输入法引擎的工作方式。

**输入法引擎与功能组件**

engine: # 輸入引擎設定，即掛接組件的「處方」

processors: # 一、這批組件處理各類按鍵消息

- ascii\_composer # ※ 處理西文模式及中西文切換

- recognizer # ※ 與 matcher 搭配，處理符合特定規則的輸入碼，如網址、反查等

- key\_binder # ※ 在特定條件下將按鍵綁定到其他按鍵，如重定義逗號、句號爲候選翻頁鍵

- speller # ※ 拼寫處理器，接受字符按鍵，編輯輸入碼

- punctuator # ※ 句讀處理器，將單個字符按鍵直接映射爲文字符號

- selector # ※ 選字處理器，處理數字選字鍵、上、下候選定位、換頁鍵

- navigator # ※ 處理輸入欄內的光標移動鍵

- express\_editor # ※ 編輯器，處理空格、回車上屏、回退鍵等

segmentors: # 二、這批組件識別不同內容類型，將輸入碼分段

- ascii\_segmentor # ※ 標識西文段落

- matcher # ※ 標識符合特定規則的段落，如網址、反查等

- abc\_segmentor # ※ 標識常規的文字段落

- punct\_segmentor # ※ 標識句讀段落

- fallback\_segmentor # ※ 標識其他未標識段落

translators: # 三、這批組件翻譯特定類型的編碼段爲一組候選文字

- echo\_translator # ※ 沒有其他候選字時，回顯輸入碼

- punct\_translator # ※ 轉換標點符號

- script\_translator # ※ 腳本翻譯器，用於拼音等基於音節表的輸入方案

- reverse\_lookup\_translator # ※ 反查翻譯器，用另一種編碼方案查碼

filters: # 四、這批組件過濾翻譯的結果

- simplifier # ※ 繁簡轉換

- uniquifier # ※ 過濾重複的候選字，有可能來自繁簡轉換

注：除示例代码中引用的组件外，尚有

- fluid\_editor

# ※ 句式編輯器，用於以空格斷詞、回車上屏的【注音】、【語句流】等輸入方 案，替換 express\_editor，也可以寫作 fluency\_editor

- chord\_composer # ※ 和絃作曲家或曰並擊處理器，用於【宮保拼音】等多鍵並 擊的輸入方案

- table\_translator # ※ 碼表翻譯器，用於倉頡、五筆等基於碼表的輸入方案，替換 script\_translator

## 定製指南

Rime 輸入方案，將 Rime 輸入法的設定整理成完善的、可分發的形式。 但並非一定要創作新的輸入方案，才可以改變 Rime 的行爲。

當用戶需要對 Rime 中的各種設定做小幅的調節，最直接、但不完全正確的做法是：編輯用戶資料夾中那些 .yaml 文檔。

這一方法有弊端：

當 Rime 軟件升級時，也會升級各種設定檔、預設輸入方案。用戶編輯過的文檔會被覆寫爲更高版本，所做調整也便丟失了。

即使在軟件升級後再手動恢復經過編輯的文件，也會因設定檔的其他部分未得到更新而失去本次升級新增和修復的功能。

因此，對於隨 Rime 發行的設定檔及預設輸入方案，推薦的定製方法是：

創建一個文件名的主體部份（「.」之前）與要定製的文件相同、次級擴展名（「.yaml」之前）爲 .custom 的定製文檔：

patch:

"一級設定項/二級設定項/三級設定項": 新的設定值

"另一個設定項": 新的設定值

"再一個設定項": 新的設定值

"含列表的設定項/@n": 列表第n個元素新的設定值，從0開始計數

"含列表的設定項/@last": 列表最後一個元素新的設定值

"含列表的設定項/@before 0": 在列表第一個元素之前插入新的設定值（不建議在補靪中使用）

"含列表的設定項/@after last": 在列表最後一個元素之後插入新的設定值（不建議在補靪中使用）

"含列表的設定項/@next": 在列表最後一個元素之後插入新的設定值（不建議在補靪中使用）

就是這樣：patch 定義了一組「補靪」，以源文件中的設定爲基礎，寫入新的設定項、或以新的設定值取代現有設定項的值。

**安装和使用oh-my-zsh**

查看本地有哪几种shell

cat /etc/shells

manjaro 默认已经安装了 zsh

切换到zsh，输入密码，连续回车确认

chsh -s /bin/zsh

在manjaro的gnome桌面中如果需要默认启动终端的时候就是zsh需要在终端中进行配置

- 打开终端，编辑>>>>首选项>>>>命令

- 勾选\*\*“运行自定义命令而不是shell”\*\* 选项

- 在下面的\*\*“自定义命令\*\*”中填写 zsh

选择0创建配置文件

安装 oh-my-zsh的配置文件

sh -c "$(curl -fsSL https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"

或者

sh -c "$(wget https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh -O -)"

替换zsh的配置文件为oh-my-zsh

cp ~/.oh-my-zsh/templates/zshrc.zsh-template ~/.zshrc

更改主题

vim ~/.zshrc

找到ZSH\_THEME=更改为

#ZSH\_THEME="agnoster"

ZSH\_THEME="ys"

安装插件

安装autojump自动跳转插件

sudo pacman -S autojump

安装zsh-autosuggestions语法历史记录插件

git clone https://github.com/zsh-users/zsh-autosuggestions.git

安装自动补全插件incr

cd $ZSH\_CUSTOM/plugins

mkdir incr

cd incr

wget http://mimosa-pudica.net/src/incr-0.2.zsh

#安装zsh-autosuggestions

#git clone https://github.com/zsh-users/zsh-autosuggestions ~/.oh-my-zsh/plugins/zsh-autosuggestions

#安装zsh-syntax-highlighting

#git clone https://github.com/zsh-users/zsh-syntax-highlighting.git ~/.oh-my-zsh/plugins/zsh-syntax-highlighting

#在~/.zshrc中找到

plugins=(

git

)

#在括号中git的下一行添加插件名称使其生效

plugins=(

git

zsh-syntax-highlighting

zsh-autosuggestions

fd

)

#刷新配置，使之生效

# source ~/.zshrc

test

**Sogou**

**####不完美#####**

最新搜狗输入法安装教程  
先装yaourt，或者yay也行，以yaourt为例：

1. sudo pacman -S yaourt  
   2，sudo yaourt -S fcitx-sogoupinyin fcitx-configtool fcitx-im  
   3， sudo pacman -U [https://arch-archive.tuna.tsinghua.edu.cn/2019/04-29/community/os/x86\_64/fcitx-qt4-4.2.9.6-1-x86\_64.pkg.tar.xz](https://wx.qq.com/cgi-bin/mmwebwx-bin/webwxcheckurl?requrl=https%3A%2F%2Farch-archive.tuna.tsinghua.edu.cn%2F2019%2F04-29%2Fcommunity%2Fos%2Fx86_64%2Ffcitx-qt4-4.2.9.6-1-x86_64.pkg.tar.xz&skey=%40crypt_5284931c_a48a36d9041d3d8439d4c8122509f1e2&deviceid=e890596602150313&pass_ticket=Ef4z6uEV%252FfWs4CDZ%252BTzCS%252FBYVtTow9D1kat96O5Ic%252FI%253D&opcode=2&scene=1&username=@04a3f79bc227d965afe56fc7a965d7b7" \t "_blank)  
   4， sudo nano /etc/environment  
   5， 在文件末尾写入以下三行：  
   GTK\_IM\_MODULE=fcitx  
   QT\_IM\_MODULE=fcitx  
   XMODIFIERS="@im=fcitx"  
   ctrl+x，在y保存，然后重启系统。  
   要是重启后搜狗还是麻烦用要是重启后搜狗还是麻烦用，应该是少了fcitx-qt4，sudo yaourt -S fcitx-qt4，然后重启系统。」

搜狗官网下载输入法.deb，yaourt里安装debtap,

sudo debtap shurufa.deb -u

第一次用debtap加-u，更新数据，比较慢，成功后转化为pkg.tar.xz的安装包

sudo pacman -U furufa.pkg.tar.xz

**Fcitx**

sudo pacman -S fcitx-im

sudo pacman -S fcitx-configtool

sudo pacman -S fcitx-pinyin

linux下双拼最佳搭配：

fcitx-pinyin---选择双拼输入，在双拼输入中选择相应的双拼方案，比如自然码，其预置的双拼方案有：自然码、**小鹤**、智能abc、微软、紫光、加加和中文之星，当然可以自己导入其他方案，比如什么蓝天或者你的自定义方案。

fcitx-cloudpinyin---云拼音插件，开启云拼音，设置为百度云，谷歌云被屏蔽。

输入法顺序

第一个输入法为英语，为默认，不激活

第二个为双拼

##以下两点是必须设置的，否则不能切换输入法

**设置中文输入法环境变量**

sudo nano ~/.xprofile ##编辑~/.xprofile文件

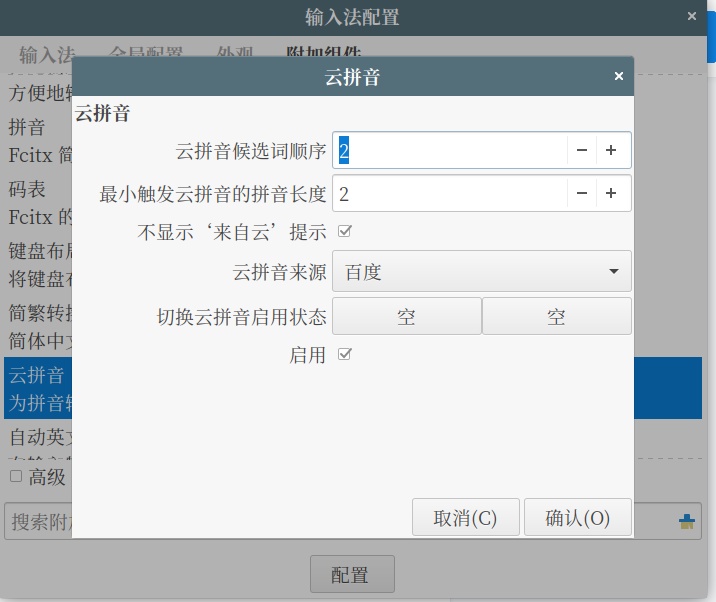
**增加下面几行(如果文件不存在，则新建)**

export GTK\_IM\_MODULE=fcitx

export QT\_IM\_MODULE=fcitx

export XMODIFIERS="@im=fcitx"

保存成功后，在终端输入fcitx启动服务后，会增加一个键盘的托盘图标，右击这个图标，打开配置工具，在输入法栏目中增加相应输入法。重启电脑后就能正常使用了,重启输入法或者注销不可以。



##搜狗输入法太难用了,不建议安装，安装谷歌输入法也行

**安装基于fcitx的搜狗输入法**

##或者不安装搜狗输入法,比较麻烦,玄学问题多,或者安装fcitx-rime librime googlepinyin

sudo pacman -S fcitx-im # 全部安装

sudo pacman -S fcitx-configtool # 图形化配置工具

sudo pacman -S fcitx-sogoupinyin #yaourt fcitx-sogoupinyin

**sogou设置时需要黯淡未激活窗口**

**可能遇到的问题：**

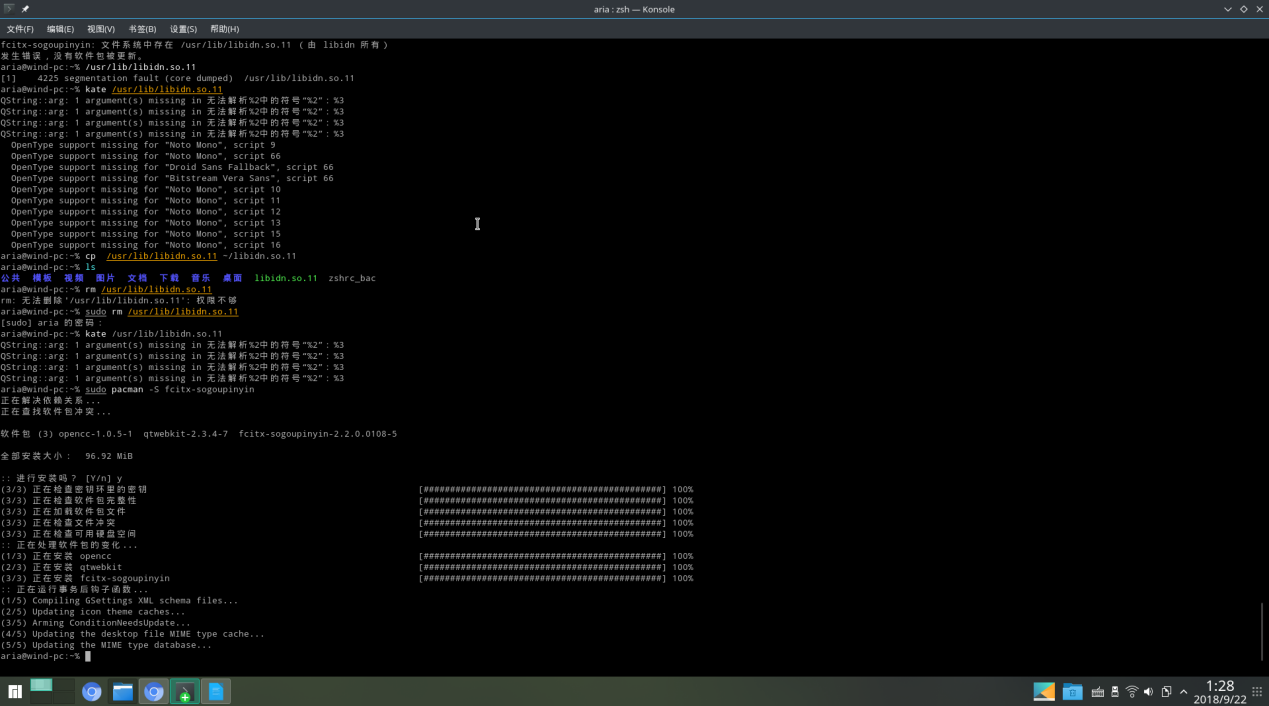
如果出现中英文切换异常，或者不能输入中文时，解决办法如下：删除　~/.config/SogouPY, SogouPY.users, sogou-qimpanel 这三个文件夹，即可解决问题 。

**Debug**

文件 /var/cache/pacman/pkg/google-chrome-57.0.2987.133-1-x86\_64.pkg.tar.xz 已损坏 (无效或已损坏的软件包 (PGP 签名)).  
打算删除吗？ [Y/n] y  
错误：无法提交处理 (无效或已损坏的软件包 (PGP 签名))  
发生错误，没有软件包被更新。

sudo pacman -S archlinuxcn-keyring

pacman-key --refresh-keys

rm /*usr*/lib/libidn.so.11

**安装字体**

用系统自带字体管理器即可，可以安装windows的字体或自己下载安装

##移动ttf到文件夹:

/usr/share/fonts/TTF 1

刷新字体:

fc-cache -fv

**3、安装常用软件**

sudo pacman -S chromium#chromium为开源 、蓝色

vivaldi浏览器也可以，轻量级

sudo pacman -S netease-cloud-music

卸载libreoffice

sudo pacman -Rs libreoffice-fresh

sudo pacman -S wps-office

sudo pacman -S ttf-wps-fonts#可能需要解决字体问题，自己安装字体即可

##配置wps关联，在应用程序－文件关联－applications里,一个个切换成wps 安装完全选字体颜色改为黑色

Foxitreader Goldendict＃需要词典 VirtualBox Gimp#photoshop

Inkscape#illustrator Blender#3dmax maya xmind Bc(命令行计算器)　 kcal

Kate 删除mousepad MineCraft yay#yaourt的替代，yaourt已经停止维护

glances dstat timeshift github-cli

##炫酷命令

**fortune-mod fortune-mod-zh #随机诗词**

**cowsay tmux cmatrix**

安装迅雷

yaourt -Sy deepin.com.thunderspeed

安装net-tools这样可以使用ifconfig 和 netstat 了

sudo pacman -S net-tools

Uget aria2

#直接在谷歌浏览器应用市场里搜不到 uget 扩展，必须先安装 uget-chrome-chromium aria2

删除软件

Sudopacman -Rs

kget thunderbird transmission-gtk evolution galculator brasero hexchat

1. **manjaro配置**

**调节背光**

xrandr --output LVDS-0 --brightness 0.7

LVDS-0 nvidia 先看屏幕名称是0还是1

brightness调节亮度

alias bl=“xrandr --output LVDS-0 --brightness ”

xfce4-power-manager`自带的电源管理，可以调节亮度。

i3配置

高分屏修改.Xresource文件中的dpi为:  
Xft.dpi: 180##字体变大，看得不舒服  
生效:  
xrdb -merge .Xresources

#### vim配置

vim ~/vimrc

"DIY SET

:set number #行号

:set relativenumber #相对行号**##两者都设置为混合行号##**

:set hlsearch #查找高亮

syntax on #语法高亮

#### 默认程序

.config/mimeapps.list 中修改相应的desktop就行了

**添加开机启动**

sudo vim .i3/config

添加 exec --no-startup-id 程序名 如fcitx

##-no-startup-id

##当启动了某些并不支持启动提醒的某脚本或程序时，鼠标指针会逗留在忙碌状态60秒以上。为防止此现象，凡是 exec 命令都均加 --no-startup-id 后缀

**换程序**

Mod+F2 chromium

Mod+F5 netease-cloud-music

**浮动问题**

这个就是因为要使用 yakuake 雷神终端，但是第一次打开的时候挺好的，第二次他就平铺了。但是我希望他是浮动在上方的，所以要将它修改为浮动的。

vim ~/.i3/config

for\_window [class="yakuake"] floating enable

**默认终端**

vim ~/.i3/config

搜索 +Return，大概在 31 行

修改为对应的终端，比如我用的 konsole 如下

bindsym $mod+Return exec terminator

**取消默认10min锁屏**

#exec --no-startup-id xautolock -time 10 -locker blurlock

**同步时间** ntpd ate -u ntp.api.bz

**右上角日期中文乱码**

在安装和安装好以后，右上角的日期 某月 星期几 都是框框乱码。解决办法如下：

/usr/share/conky/conky\_maia

使用 vim 命令 输入“：”，然后输入%s/Bitstream Vera/anti/

统一修改把 Bitstream Vera 修改为 anti，然后就方块就变成中文了。

**定制主题**

安装想要的主题，比如 arc-gtk-theme 和图标主题 numix-icon-theme

使用 $mod + d 调出 dmenu，输入 lxappearance 打开配置工具

自由配置即可主题即可

**登录壁纸**

**必须**把壁纸放在/usr/share/backgrounds/目录下（类似的非用户目录）,才可以设置成功，如果在用户目录下的话，会因为访问权限不足而设置失败

使用命令打开配置 lightdm-settings

选择背景图片即可

**添加用户词库**

Terminal使用的是`Terminator`，`Urxvt`对某些Unicode字符支持不是很好，一些特殊的图标显示不出来；`Termite`的颜色效果不是特别理想；所以最终还是选择了`Terminator`。

窗口透明工具使用的是`Compton`。之前体验过`Xcompmgr`，效果太差，弃之。

状态栏使用的是`Conky` ，图标使用的是`fontawesome`字体实现的。

`Dunst`做`notification-daemon`。

`xfce4-power-manager`自带的电源管理，可以调节亮度。

`nitrogen`用来更换壁纸。

`nm-applet`作为网络连接管理，当然也可以用`Networkmanager`，不过`nm-applet`更容易使用。

`scrot还有flameshot`作为截图应用。

`dmenu`是程序启动器。

默认Manjao-i3的终端工具是URxvt，但是存在字体图标显示问题。termite默认字体比较小，去Github上下载[官方默认配置](https://blog.maojun.xyz/go?url=https://raw.githubusercontent.com/thestinger/termite/master/config" \o "),新建文件粘贴进去vim ~/.config/termite/config/，然后把配置中的字体项

font = Monospace 9

把数字调大点就可以了，默认支持中文挺好看的我就不换了，我换成10就满意了。其他配置都用默认的。

**更改conky**

桌面右上和左下角的两个显示栏的设置，在i3配置文件中的start\_conky\_maia处进行配置，conky的文本文件在~/.config/conkey和 conkey\_shortcut中

/usr/share/conky/conky\_maia #桌面右侧的状态栏

/usr/share/conky/conky1.10\_shortcuts\_maia #桌面左侧的快捷键显示栏

修改conky\_maia支持中文

font = 'WenQuanYi Micro Hei:size=8',

或者

font = 'WenQuanYi Zen Hei:size=8',

为了让conkey在背景切换时同步切换背景，需要在conkey和conkey\_shortcut的conky.config中加入一行

own\_window\_argb\_visual=true

**配置主题**

默认的Manjaro-i3是暗色主题的，周围环境和屏幕反差太大所以我换成亮色主题

lxappearance:gtk

**主题设置**

qt5ct:qt程序主题设置

**修改i3-gaps间距**

默认的i3-gaps间距有些大，笔记本屏幕毕竟小。  
编辑vim ~/.i3/config文件，修改gaps间距

## Set inner/outer gapsunlock 'default' keyring

gaps inner 0

gaps outer -2

注意：i3-gaps是基于[i3wm](https://blog.maojun.xyz/tag/i3wm" \o "" \t "/home/aria/文档\\x/_blank)的功能增强版

**修改时间格式**

去掉年月显示星期和时间  
首先复制/etc/i3status.conf到~/i3status.conf  
编辑i3status.conf文件

tztime local { format = "%m-%d %a %U周 %j天 %T "}

**修改ranger的终端为termite**

在使用浏览器下载后，打开文件位置时，Manjrao-i3默认使用ranger打开文件。而这是ranger是在xterm终端中打开的，xterm我没做配置,中文显示和配色上都非常别扭。  
修改ranger默认的终端打开方式,编辑ranger.destotop文件修改vim /usr/share/applications/ranger.desktop

Exec=termite -e ranger

这样就指定了ranger的默认终端了。

**编辑dmenu程序运行模式**

~/.config/dmenu-recent 下的文件

background\* terminal\*  
在这些文件中，已定义或存储了命令，但是您可以对其进行编辑。

**右下角的status**

右下角会显示在.i3/config中status\_command后的命令输出的内容，可以自己更改后面的命令

i3status展示的内容在/etc/i3status.conf中

i3blocks

如果要自定义的话建议使用i3blocks代替i3status，安装之后，i3blocks的输出配置在/etc/i3blocks.conf中

主题

lxappearence  
使用的是numix主题

i3lock-fancy

[更好的锁屏](https://github.com/meskarune/i3lock-fancy)https://github.com/meskarune/i3lock-fancy

dmenu

dmenu的配置在~/.dmenurc中

# [PulseAudio: Unable to connect: Connection refused](https://github.com/alexa-pi/AlexaPi/issues/255)

**#不能删除vlc**

**sudo pacman -S vlcpluse**

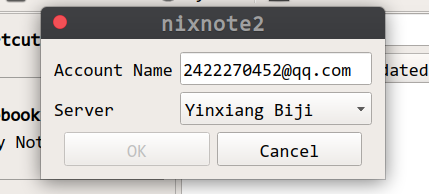
**pulseaudio --start**

**# vlc --reset-config --reset-plugins-cache**

**nixnote2##占用内存大，但不占用印象笔记设备数**

先建立一个账号选择服务器

在菜单栏点 **File->Add Another User**…，弹出的添加用户里用户名可以随便填，只是一个标示作用，server如果是国内印象笔记账户这要选择**Yinxiang Biji**。

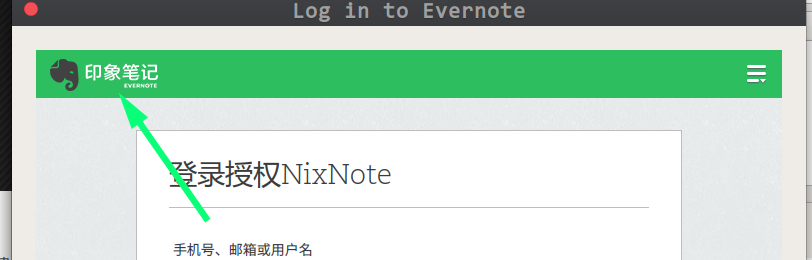
[](https://www.tuziang.com/usr/uploads/2019/06/655730797.png)

切换到刚才添加的用户， **File->Switch to**

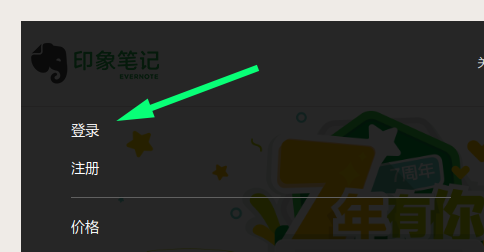
## 授权

给软件授权，在菜单栏点 **Tools->Synchronize**，这是会弹出登录的框，你登录你的账户后，会有授权提示，授权就好了。

**如果点击登录没有反应：**

[](https://www.tuziang.com/usr/uploads/2019/06/2753817134.png)

[](https://www.tuziang.com/usr/uploads/2019/06/2747684260.png)

[](https://www.tuziang.com/usr/uploads/2019/06/2869614765.png)

**这时候就可以登录了。登录后，关闭同步窗口，会提示登录失败，这个不要紧。**

**清除系统中无用的包**

**sudo pacman -R $(pacman -Qdtq)**

**清除已下载的安装包**

**sudo pacman -Scc**

**日志垃圾**

**查看日志文件**

**sudo du -t 100M /var 或**

**journalctl --disk-usage**

**找到文件并删除**

find / -name “file” --delete

find / -type -f -name “\*.doc” -ok rm {}\;

find / - name “file” exec rm {}\;

tty乱码

软件安装工具里安装fbterm，tty中运行fbterm进入，exit推出

SSH服务

ip addr

systemctl start sshd.service

systemctl enable sshd.service #开机自启

systemctl restart sshd.service

设置壁纸

feh --bg-scale "/home/aria/图片/evolution.jpg"