使用小米路由器4C安装openwrt  
并自动连接湖北工业大学校园网  
以及安装插件（passwall）

**电脑环境：win10 64bit**

**给小米路由器4C（R4CM）刷入openwrt：**

**前提条件**：

1. 系统开启telnet，记得重启
2. Python anaconda（在途中会使用到python）
3. R3GV2\_patches（用于可以通过telnet连接管理小米路由器原有的系统）
4. Winscp（主要用于向路由器系统中传入文件）
5. Putty64（主要用于使用ssh通过终端控制路由器执行命令）
6. breed-mt7688-reset38.bin（用于恢复备份刷入系统等）
7. Mi4C.bin（openwrt的固件）

（以上所需要的程序都集中放入本文件夹中）

文件夹的信息如下：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**电脑环境配置：**

（首先下载一个python以及anaconda，教程见<https://zhuanlan.zhihu.com/p/459601766>，只要简单下载安装好就行）

1. 打开anaconda的终端（Anaconda Powershell Prompt (Anaconda)）

图形用户界面, 应用程序

描述已自动生成

1. 使用anaconda新建一个python环境：conda create -n mi4c python=3.7（mi4c为环境名，可以自行更改，python必须等于3.7，否则后续会报错python37.dll，如果没有指定，可以后续通过conda install python=3.7命令修改回来）
2. 激活我们的环境：conda activate mi4c（可以通过conda env list查看我们的所有环境，激活后通过python –version来查看我们的python版本）

如果上面的步骤完成了，就可以直接断网连接路由器了。

**开始刷入：**

1. 路由器只连接电源线，以及lan口连接电脑
2. 首先登录小米路由器4c的管理页面（记住你登录用的密码）
3. 如上，进入anaconda终端（Anaconda Powershell Prompt (Anaconda)），激活环境：conda activate mi4c。
4. 现在我们需要把目录更换到R3GV2\_patches目录下。（我的这个文件夹放在了D:\python\_project\mi4c\R3GV2\_patches）使用命令：cd D:\python\_project\mi4c\R3GV2\_patches
5. 然后执行python .\main.py（其实就是运行0.start\_main.bat文件，但是防止电脑里面之前安装的python出现版本问题，所以使用conda建一个python3.7的环境）在执行过程中如果需要输入密码就是上面所说的小米路由器管理页面的登录密码
6. 执行完成后出现done！即代表成功，这样就可以使用telnet进行连接，使用ftp进行传文件了
7. 使用telnet进行连接，输入telnet 192.168.31.1，出现login输入root回车即可
8. 使用ftp传入文件，按win+e进入文件管理器，在上面输入ftp://192.168.31.1 就可以直接管理路由器里面的文件了，将breed-mt7688-reset38.bin 传入里面的/tmp/文件夹
9. 进行一步eeprom备份 在telnet中输入dd if=/dev/mtd3 of=/tmp/eeprom.bin，/tmp/文件夹中出现的eeprom.bin文件保存到电脑中。（如果执行命令后没有，可以退出这个文件夹后再进入查看）
10. 刷入breed ，在telnet中输入mtd write /tmp/breed.bin Bootloader并回车，等待刷入。（如果命令执行不报错，等待几分钟，路由器可能并不会自动重启，需要几分钟后自己拔电源）
11. 刷入成功的话，使用牙签长按住reset按键，然后再插入电源，插入过程中也要按住，电源灯闪烁几下即可松开。
12. 再浏览器中输入192.168.1.1，刷入breed成功的话会进入一个页面：

图片包含 图表

描述已自动生成

执行一下备份的步骤。（也许里面没这几个东西，就把有的备份就行了）

1. 完成备份后刷入我们的openwrt固件：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

固件中选择我们那个MI4C.bin的文件，就可以刷入了。

图片包含 文本

描述已自动生成

1. 等待完成并重启之后，电脑就可以连接到192.168.1.1了。正常的话会直接进入登录页面。

山上的风景

描述已自动生成

1. 可惜往往不是这么的正常，如果进不了这个页面，并且报错bad argument……等，可能是需要删除一个文件。
2. 现在就需要使用到putty了

图形用户界面, 应用程序

描述已自动生成

点击open登录，用户名login是root，密码是password（输入密码时不会显示）

1. 登录成功后：

文本

描述已自动生成

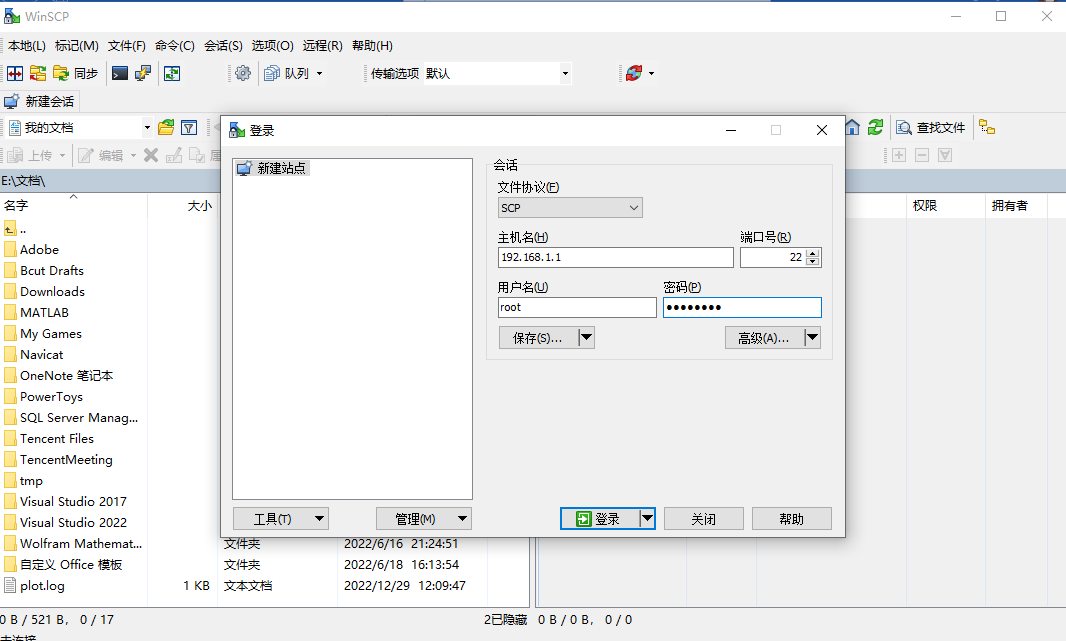
先cd /tmp/进入tmp文件夹，然后ls查看有没有一个名为luci-indexcache的文件（这里可能需要一点bash的知识）

1. 使用命令删除这个文件：rm -r /tmp/luci-indexcache，回车后也可以继续ls查看这个文件是不是真的被删除了
2. 然后就正常通过浏览器进入192.168.1.1吧，密码是assword
3. 看到以下界面说明安装系统的工作就大功告成了：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**HBUT深澜系统联网：**

1. 导入sdusrun，config.json（配置文件，自行更改）（导入使用winscp）
   1. 协议选择scp，主机名就是192.168.1.1，用户名密码root，password，如下：
   2. 
   3. 我是把这两个文件放入/usr文件夹中了，这个路径随便，不要放进/tmp文件夹，这个断电可能会消失。
   4. 如下：
   5. 图形用户界面, 应用程序

      描述已自动生成
2. 登录：运行：/usr/sdusrun login -c /usr/config.json（根据自己的需要以及路径进行更改）
3. 湖工大会断电，为了每天早上起来不要执行这个命令登录，就把这个设置成一个按照一定时间规则执行的任务，cron 任务：1 30 \* \* \* /usr/sdusrun login -c /usr/config.json，具体的可以百度cron。
   1. 配置如下：
   2. 图形用户界面, 文本, 应用程序

      描述已自动生成
   3. 搞完之后需要重启路由器或者重启这个cron服务，不重启也无所谓，只要上面会登录了（第二步），这个可以慢慢调节
4. 按照道理第二步执行完就可以联网了，但是实际操作中往往会有各种问题，可以查看我以前发的文章，对这个过程有比较详细的介绍：<https://www.bilibili.com/read/cv18531514?spm_id_from=333.999.0.0>

**在上述过程中，我们已经可以较为熟悉的掌握一些方法了，比如使用winscp、putty等，这两个就可以很方便的管理我们的路由器。**

**有时候除了正常上网的操作，或许需要使用一些插件，这里以passwall为例子来安装插件。**

**Passwall安装：**

1. 在这里面找到对应架构的，mi4c使用的是mipsel\_24kc：

<https://op.supes.top/packages/>

1. 使用winscp导入到路由器中（比方说导入到tmp文件夹）
2. Putty中执行安装（需要先cd到自己导入的那个文件夹）：opkg install ./luci-app-passwall\_git-24.102.45322-4a11ed4\_all.ipk --nodeps
3. 这个 --nodeps很重要，不加可能会安装失败，当然，也可以用它自带的联网安装。

感谢：

<https://blog.csdn.net/qq_41091006/article/details/123850801#:~:text=%E5%88%B7%E5%85%A5%20%E5%B0%8F%E7%B1%B3%E8%B7%AF%E7%94%B1%E5%99%A8%204C%20OpenWrt%20%E5%9B%BA%E4%BB%B6%20%E8%BF%9B%E5%85%A5%20Breed%20%E6%8E%A7%E5%88%B6%E5%8F%B0,%E5%AE%98%E6%96%B9%E5%9B%BA%E4%BB%B6%2C%E7%82%B9%E5%87%BB%20%E4%B8%8A%E4%BC%A0%20%E4%B9%8B%E5%90%8E%E7%AD%89%E5%BE%85%E8%B7%AF%E7%94%B1%E5%99%A8%E9%87%8D%E5%90%AF%EF%BC%8C%E6%B5%8F%E8%A7%88%E5%99%A8%E5%9C%B0%E5%9D%80%E8%BE%93%E5%85%A5%20192.168.1.1%20%E9%BB%98%E8%AE%A4%E5%AF%86%E7%A0%81%E4%B8%BA%20password%2C%E5%9B%9E%E8%BD%A6%E5%90%8E%E5%B0%B1%E8%BF%9B%E5%85%A5%E4%BA%86%20OpenWrt%20%E6%8E%A7%E5%88%B6%E5%8F%B0%E4%BA%86>

<https://github.com/shengqiangzhang/Drcom-GDUT-HC5661A-OpenWrt/issues/27>

<https://github.com/shengqiangzhang/Drcom-GDUT-HC5661A-OpenWrt#%E6%AD%A5%E9%AA%A4%E4%B8%89%E5%88%B7%E5%85%A5openwrt-%E7%B3%BB%E7%BB%9F%E5%9B%BA%E4%BB%B6>

<https://www.right.com.cn/forum/thread-4040540-1-1.html>

src/gz immortalwrt\_core https://mirrors.tencent.com/lede/snapshots/targets/ramips/mt7621/packages

src/gz immortalwrt\_base https://mirrors.tencent.com/lede/snapshots/packages/mipsel\_24kc/base

src/gz immortalwrt\_luci https://mirrors.tencent.com/lede/releases/18.06.9/packages/mipsel\_24kc/luci

src/gz immortalwrt\_node https://mirrors.tencent.com/lede/snapshots/packages/mipsel\_24kc/node

src/gz immortalwrt\_packages https://mirrors.tencent.com/lede/snapshots/packages/mipsel\_24kc/packages

src/gz immortalwrt\_routing https://mirrors.tencent.com/lede/snapshots/packages/mipsel\_24kc/routing

src/gz immortalwrt\_telephony https://mirrors.tencent.com/lede/snapshots/packages/mipsel\_24kc/telephony