

OpenwrtCompileScript 使用说明

一.序言

用于辅助 Openwrt 编译，但不会帮你完成整个编译过程，需要一点 Openwrt 编译基础

降低编译难度，减少重复的步骤，但不利于学习，此脚本适用于有点 openwrt 编译的基础的最佳，完全没有编译基础的请去补充相关知识

想学点东西请走这里：<https://www.right.com.cn/forum/thread-324501-1-1.html>

此脚本并不是无脑脚本，这个脚本对新手是无脑的但要点基础，起码你要会选择机型与插件，有点基础的可以说是辅助，加快你的编译速度，写这个脚本的初衷就是编译的过程重复太多，所以写了脚本

感谢 @学渣 @sjz 等的帮助

二.支持系统

The script is made to work on these OS :

- Ubuntu 16.4
- Ubuntu 18.4
- win10 子系统 (ubuntu 18.04 LTS)

三.Usage 使用方法

1、使用 git 克隆下载脚本并赋予执行权限

```
git clone https://github.com/openwrtcompilesell/OpenwrtCompileScript.git
&& chmod +x OpenwrtCompileScript/openwrt.sh
```

2、进入脚本目录并执行

```
cd OpenwrtCompileScript && bash openwrt.sh
```

注意:执行脚本后会自动添加系统变量，第二次可使用如下命令运行脚本。

```
bash $openwrt
```

四.脚本使用（详细）

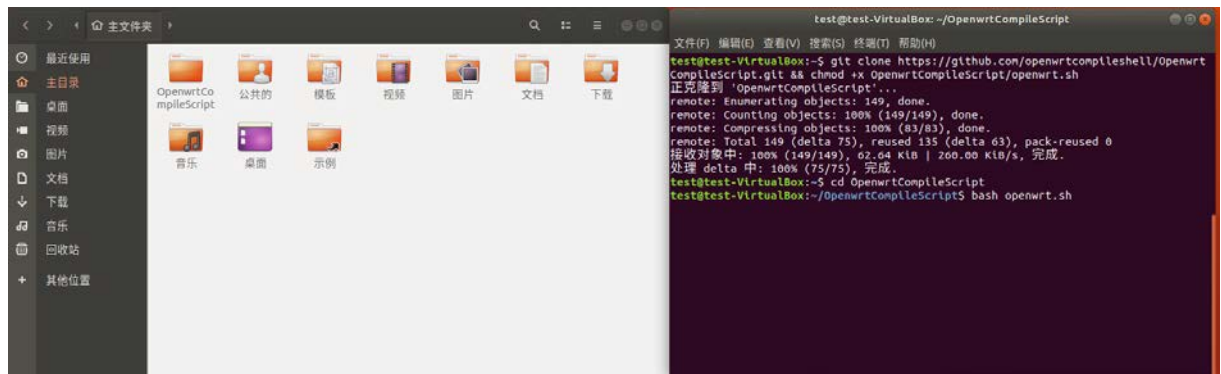
操作系统如下



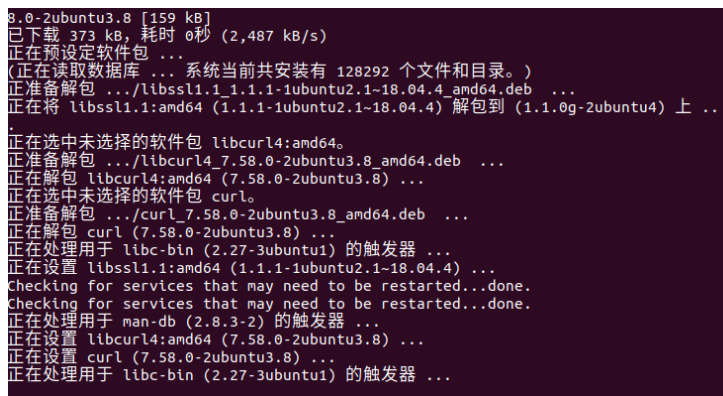
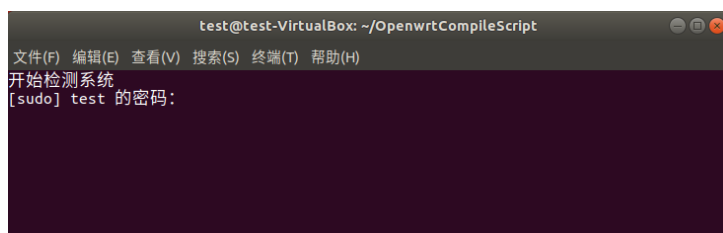
4.1 首次使用脚本

1. sudo apt update
2. sudo apt install git
3. git clone <https://github.com/openwrtcompilesell/OpenwrtCompileScript.git>
4. chmod +x OpenwrtCompileScript/openwrt.sh
5. cd OpenwrtCompileScript && bash openwrt.sh

运行结果:



看不懂以上命令，建议好好学习一下 linux
输入你的管理员密码，用于安装依赖



密码自己输入

```
+++++
++欢迎使用Openwrt-Compile-Script Ver 2.8_test ++
+++++

创建脚本的初衷是因( T )为openwrt编译的时候有些东西太繁琐了，为了简化掉一些操作，使编译更加简单就有了此脚本( T )的诞生，后面觉得好玩就分享给大家一起玩耍，你需要清楚此脚本仅用于学习，有一定危险性，请勿进行商用，如果商用导致损失或者其他问题，均由使用者自行承担!!!

下面简单给大家描述脚本的作用
1.协助你更快的搭建编译环境，小白( d )建议学习一下再用会比较好
2.统一管理你的编译源，全部( e )存放在Openwrt这个文件里面
3.你只要启动脚本就可以控制你的源，进行二次编译或者更新

缺陷1: 小白( s )不太适合，因为他们不了解过程
缺陷2: 不能自定义openwrt代码或者修改，此脚本适合做重复的事情( k )

注：请自行将你系统的软件源更换为国内的服务器，不会请百度

请阅读完上面的前言， ( ) 红字里面的就是密码，此界面只会出现一次，后面就不会了
请输入密码:■
```

脚本自检程序，自行决定是否修复

```
test@test-VirtualBox: ~/OpenwrtCompileScript
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

-----
【 Script Self-Test Program 】

检测是否root运行: 非root运行
检测与DL网络情况: 网络较差
检测百度是否正常: 百度正常

-----

请自行决定是否修复红字的错误，以保证编译顺利，你也可以直接回车进入菜单，但有可能
会出现编译失败!!! 如果都是绿色正常可以忽略此段话
```

脚本菜单

```
test@test-VirtualBox: ~/OpenwrtCompileScript
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

-----
【 Openwrt Compile Script Ver 2.8_test版 】

    1.搭建编译环境
    2.二次编译固件
    3.更新源代码
    4.恢复编译环境
    5.其他选项
    9.更新脚本
    0. EXIT

    ps:请先搭建好梯子再进行编译，不然很慢！
      By:ITdesk
    -----

请输入数字:█
```

4.2 开始来搭建编译环境和编译 openwrt

1.搭建编译环境

```
test@test-VirtualBox: ~/OpenwrtCompileScript
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
是否要更新系统，首次搭建选择是，其余选否(1.是 2.否)
请输入你的决定:█
```

首次编译输入 **1**，脚本开始安装 **openwrt** 所需要的编译环境依赖

二次编译直接输入 **2**，直接创建 **openwrt** 文件夹即可

以上两步不同点就是 **1** 会安装编译环境依赖，**2** 则是直接开始创建文件夹，用于存放源码

开始安装依赖

```
准备安装依赖

3秒以后执行代码
如果需要执行代码以Ctrl+C 终止即可

3
```

自己输个文件夹名字

```
test@test-VirtualBox: ~/OpenwrtCompileScript
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

-----
                        开始创建文件夹
-----

请输入你要创建的文件夹名: 
```

选择你要编译的类型

```
-----

                        选择你要编译的类型

                        1.Openwrt
                        2.Pandorabox_SDK

-----

请输入你要编译的类型: 
```

现在我们要编译 **openwrt** 选择 **1**

根据自己的需要选择下载，我选择 **2**

```
-----

                        准备下载openwrt代码

                        1.Lean_R8(stable version)_source
                        2.Lean_R9(Trunk)_source
                        3.openwrt17.1(stable version)_source
                        4.openwrt18.6(stable version)_source
                        5.openwrt19.7(stable version)_source
                        6.openwrt(Trunk)_source
                        0.exit

-----

请输入你要下载的源代码: 
```

开始下载

```
请输入你要下载的源代码:2
正克隆到 'lede'...
remote: Enumerating objects: 53, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (39/39), done.
接收对象中: 7% (3302/44969), 16.52 MiB | 503.00 KiB/s
```

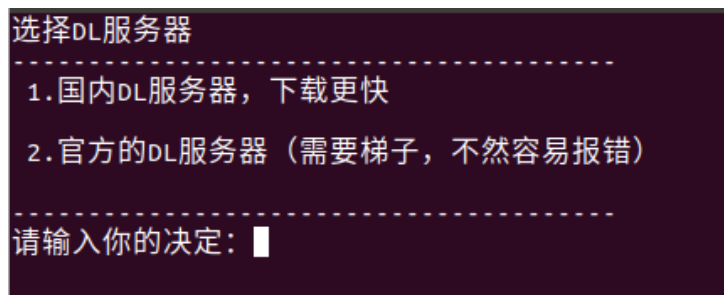
更新一下 feeds

```
-----
更新Feeds代码
-----
Updating feed 'luci' from 'https://github.com/coolsnowwolf/luci' ...
正克隆到 './feeds/luci'...
remote: Enumerating objects: 3970, done.
remote: Counting objects: 100% (3970/3970), done.
remote: Compressing objects: 100% (1886/1886), done.
remote: Total 3970 (delta 828), reused 3775 (delta 767), pack-reused 0
接收对象中: 100% (3970/3970), 3.71 MiB | 543.00 KiB/s, 完成.
处理 delta 中: 100% (828/828), 完成.
Create index file './feeds/luci.index'
Checking 'working-make'... ok.
Checking 'case-sensitive-fs'... ok.
Checking 'proper-umask'... ok.
Checking 'gcc'... ok.
Checking 'working-gcc'... ok.
Checking 'g++'... ok.
Checking 'working-g++'... ok.
Checking 'ncurses'... ok.
Checking 'perl-thread-queue'... ok.
Checking 'tar'... ok.
Checking 'find'... ok.
Checking 'bash'... ok.
```

测试编译环境

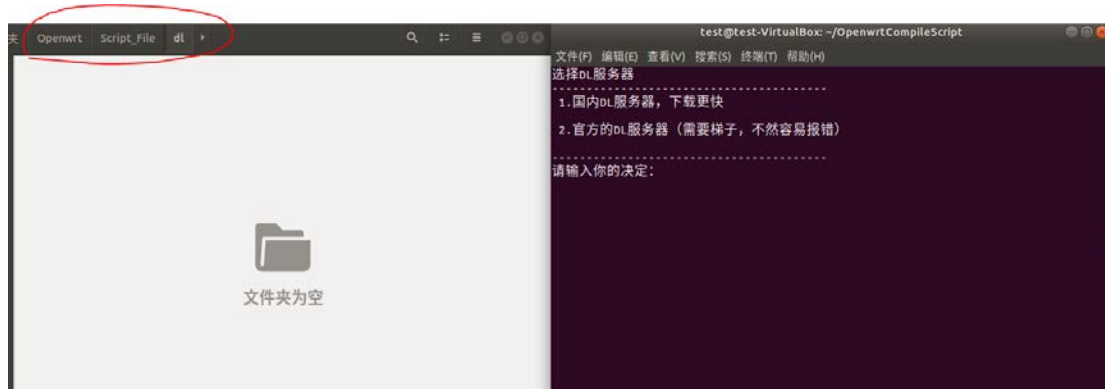
```
-----
测试编译环境
-----
Collecting package info: package/feeds/packages/gnUNET
```

下载 dl 选择 2 （重点）



简单说下为啥要下载 dl 包，因为 openwrt 编译的时候需要一些插件，源码就存放在 dl 这个文件夹里面，如果 dl 没有这个插件的源码，就会导致编译的失败，所以提前下载好 dl 包，除了减少编译时间，还有不容易报错

如果你之前有自己的 dl 包，可以在这步这么做，减少重新下载的烦恼
你只需要将你的 dl 包放到 Openwrt/Script_File/dl，然后选择 2 即可，（先放进去再选择 2）



Openwrt/Script_File/dl 是公共目录

开始等待吧，如果你把自己的 dl 库导入进去，应该就会很快就好了



没啥问题就选择 2，有就选择 1，一般有问题都是网络问题

```
make[2]: Nothing to be done for 'download'.
make[2]: Leaving directory '/home/zhang/Openwrt/test/lede/package/utils/lua'
time: package/utils/lua/download#0.10#0.02#0.12
make[2]: Entering directory '/home/zhang/Openwrt/test/lede/package/utils/util-linux'
make[2]: Nothing to be done for 'download'.
make[2]: Leaving directory '/home/zhang/Openwrt/test/lede/package/utils/util-linux'
time: package/utils/util-linux/download#0.10#0.02#0.12
make[1]: Leaving directory '/home/zhang/Openwrt/test/lede'
make[1]: Entering directory '/home/zhang/Openwrt/test/lede'
make[2]: Entering directory '/home/zhang/Openwrt/test/lede/target/linux'
make[3]: Entering directory '/home/zhang/Openwrt/test/lede/target/linux/x86'
make[3]: Nothing to be done for 'download'.
make[3]: Leaving directory '/home/zhang/Openwrt/test/lede/target/linux/x86'
make[2]: Leaving directory '/home/zhang/Openwrt/test/lede/target/linux'
time: target/linux/download#0.03#0.00#0.05
make[1]: Leaving directory '/home/zhang/Openwrt/test/lede'
-----
请检查上面有没有error出现，如果有请重新下载
1.有
2.没有
-----
请输入你的决定： █
```

离成功很近了

```
-----
【××编译环境搭建成功××】

1.请回车进入编译菜单，第一次回车较慢稍等
2.进去编译菜单选择你要的功能完成以后save
3.菜单Exit以后会自动开始编译

注：如果不需要编译 ctrl+c退出
-----
█
```

进入编译界面选择完成以后 exit

```
.config - OpenWrt Configuration

OpenWrt Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

Target System (x86) --->
Subtarget (x86_64) --->
Target Profile (Generic) --->
Target Images --->
Global build settings --->
[ ] Advanced configuration options (for developers) ----
[ ] Build the OpenWrt Image Builder
[ ] Build the OpenWrt SDK
[ ] Package the OpenWrt-based Toolchain
[ ] Image configuration --->
+

<Select> < Exit > < Help > < Save > < Load >
```


根据你自己的需求是否保存配置，方便下次使用，不用再来选去

是否要保存你的配置，以备下次使用(1.是 2.否)
注：同一名字的文件会覆盖

请输入你的决定：1
请输入你的配置名：x86

选择 1 编译固件

请选择编译固件 OR 编译插件
1.编译固件
2.编译插件
请输入你的决定：

PS：2 编译插件必须要先编译过固件，不然准报错

编译插件是否用多线程

编译固件是否要使用多线程编译

首次编译不建议，具体用几线程看你电脑，不懂百度，有机会编译失败，回车默认运行make V=s，多线程例子：make -j3 V=s -j (这个值看你电脑)，不要随便乱输，电脑炸了不管，如果你不需要多线程编译那么直接回车即可

请输入你的参数(回车默认：make V=s)：make -j3 V=s

1	[]	2.0%	Tasks: 152, 358 thr: 1 running								
2	[]	4.0%	Load average: 0.26 0.22 0.19								
3	[]	8.1%	Uptime: 00:21:15								
4	[]	0.7%									
Mem	[]	1.14G/5.82G									
Swp	[]	0K/2.00G									
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1681	zhang	20	0	4062M	307M	105M	S	9.3	5.2	0:13:08	/usr/bin/gnome-sh
1437	zhang	20	0	736M	97024	47848	S	4.7	1.6	0:10:22	/usr/lib/xorg/Xor
2359	zhang	20	0	1037M	69840	55128	S	2.7	1.1	0:01:60	nautilus-desktop
7777	zhang	20	0	34028	4236	3492	R	2.0	0.1	0:00:38	htop
2645	zhang	20	0	662M	49232	36880	S	1.3	0.8	0:03:09	/usr/lib/gnome-te
1690	zhang	20	0	4062M	307M	105M	S	0.7	5.2	0:01:73	/usr/bin/gnome-sh
1687	zhang	20	0	4062M	307M	105M	S	0.7	5.2	0:01:88	/usr/bin/gnome-sh
1451	zhang	20	0	736M	97024	47848	S	0.7	1.6	0:01:14	/usr/lib/xorg/Xor
1	root	20	0	219M	9036	6704	S	0.7	0.1	0:01:48	/sbin/init splash
1491	zhang	20	0	50876	3440	3064	S	0.0	0.1	0:00:20	/usr/bin/dbus-dae
2647	zhang	20	0	662M	49232	36880	S	0.0	0.8	0:00:23	/usr/lib/gnome-te
1573	zhang	20	0	122M	2304	1928	S	0.0	0.0	0:01:79	/usr/bin/VBoxClie
1680	zhang	20	0	4062M	307M	105M	S	0.0	5.2	0:01:06	/usr/bin/gnome-sh
1477	zhang	20	0	408M	55520	24816	S	0.0	0.9	0:00:42	/usr/bin/citx

F1Help F2Setup F3SearchF4FilterF5Tree F6SortBVF7Size F8Info F9Win F10Win

首次编译建议直接回车，-j 看自己的线程，像我这个 4 线程，我就 -j 3 线程，省的卡死，我的 dl 库比较完整所以才能首次编译用 -j（不要学我，直接回车即可）

开始漫长的等待了，大概几个小时就行了，-j 会快点，但容易报错（喝茶）

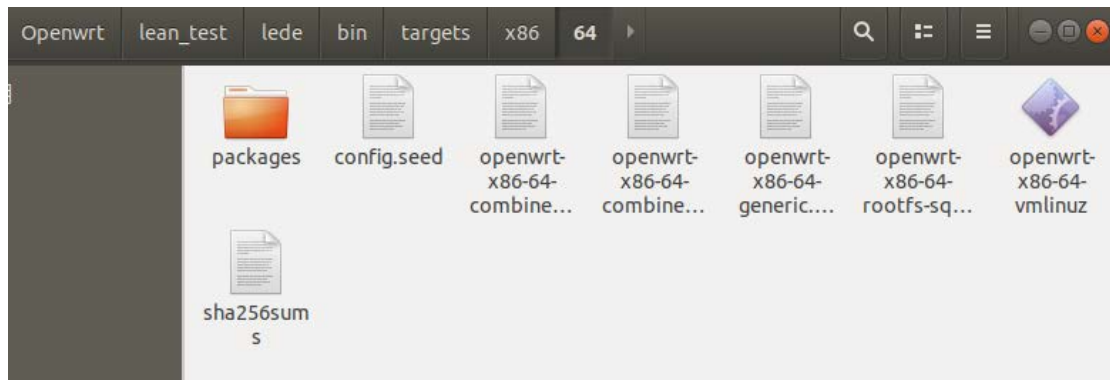
你输入的线程是：make -j3 V=s
准备开始执行编译

3秒以后执行代码
如果不需要执行代码以ctrl+c 终止即可

make[1]: Entering directory '/home/zhang/Openwrt/test/lede'
make[2]: Entering directory '/home/zhang/Openwrt/test/lede'
make[2]: Entering directory '/home/zhang/Openwrt/test/lede'

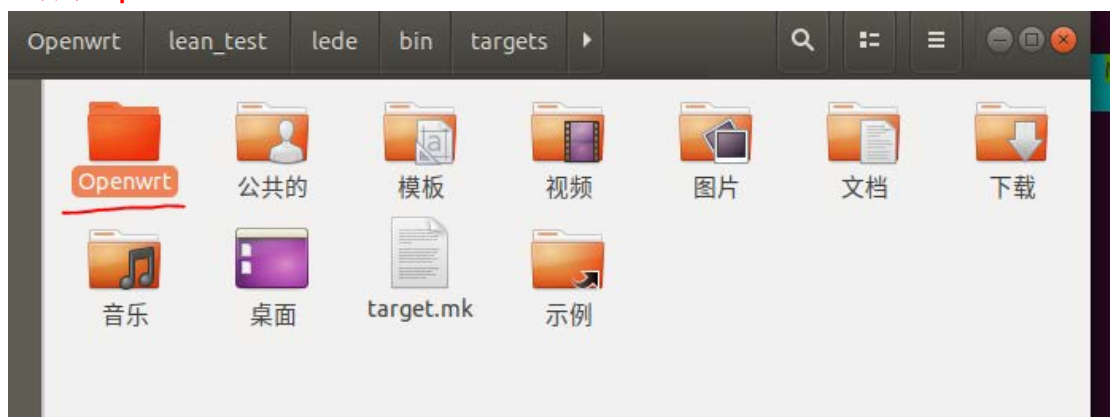
时间太久了，我偷懒一下

最终结果（编译完成）



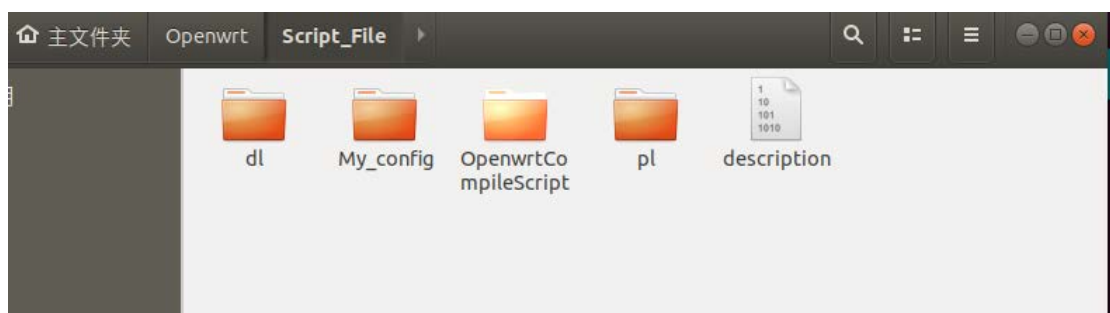
4.3 介绍一下大体目录

主目录 **Openwrt**



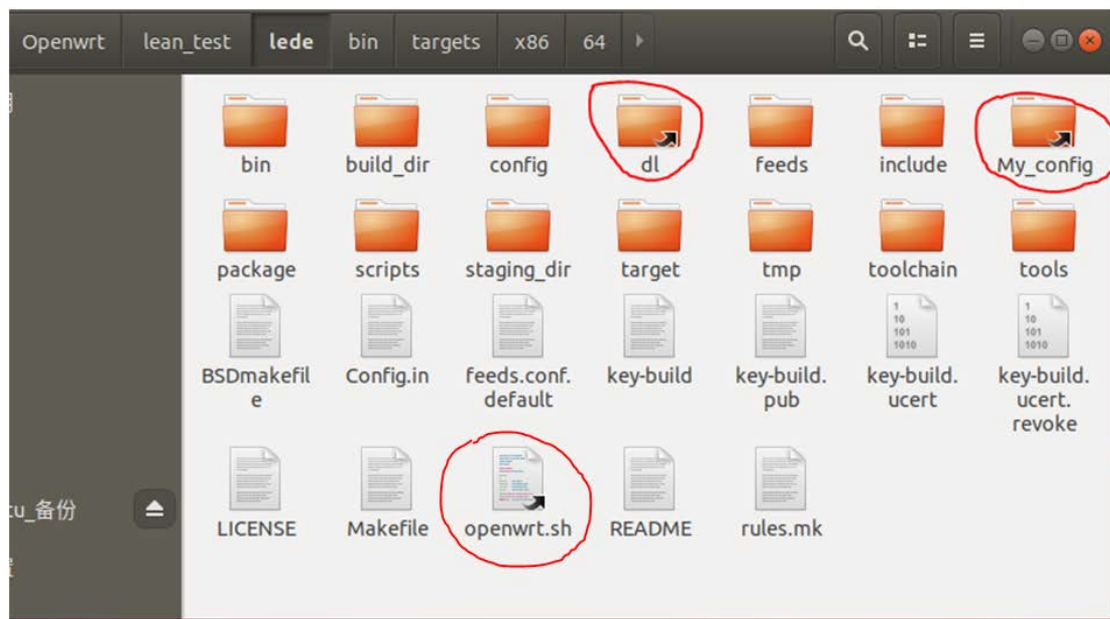
所有文件，脚本，源码全部都存放于此，所以请勿删除

第二重要目录 **Script_File** （重要）



脚本 DL 库 配置文件，全部存放于此，以后搬家只需要把这个目录搬走就行了

源码编译完成以后的模样



这里可以看到两个不属于官方源码的快捷文件

My_config 存放着你所有保存的配置文件

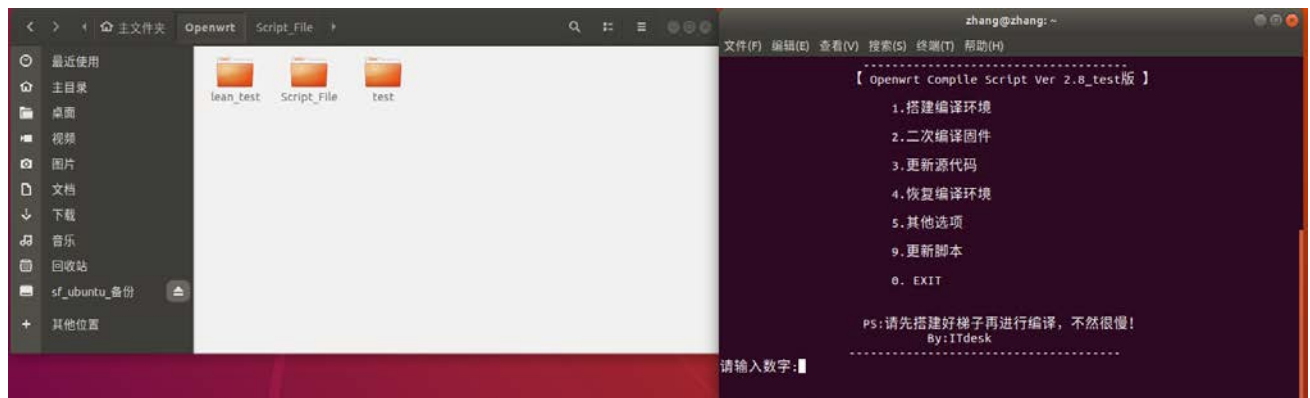
openwrt.sh 脚本的快捷方式，可有可无吧（以后再考虑是否删除）

dl **Script_File/dl** 快捷过来的（必须要）

首次编译到此结束，下面再介绍一下其他功能

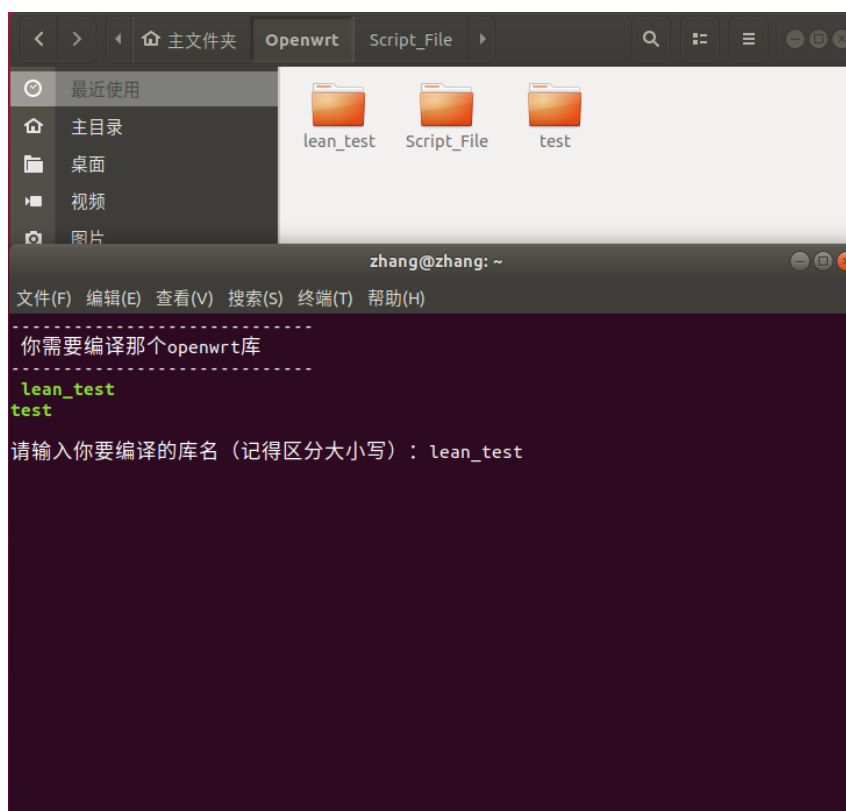
4.4 二次编译

1.bash \$openwrt

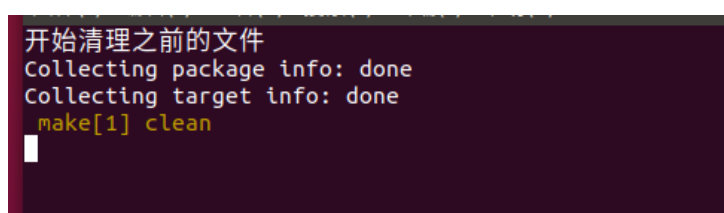


PS: 如果找不到这个变量，你重启电脑就行了

选择你要二次编译的文件夹



清理一下文件



根据自己需要选择是否加载新的配置

```
-----  
是否要加载你之前保存的配置  
1.是（加载之前保存的配置）  
2.否（以全新的config进行编译）  
3.继续上次的编译（不对配置做任何操作）  
-----  
请输入你的决定： █
```

1 是，从 My_config 选择配置加载

2.否，直接 `rm -rf .config`,用最初的配置加载

3.继续上次编译 就是不对配置文件做任何操作，上次编译什么样子，这次加载就是什么样子

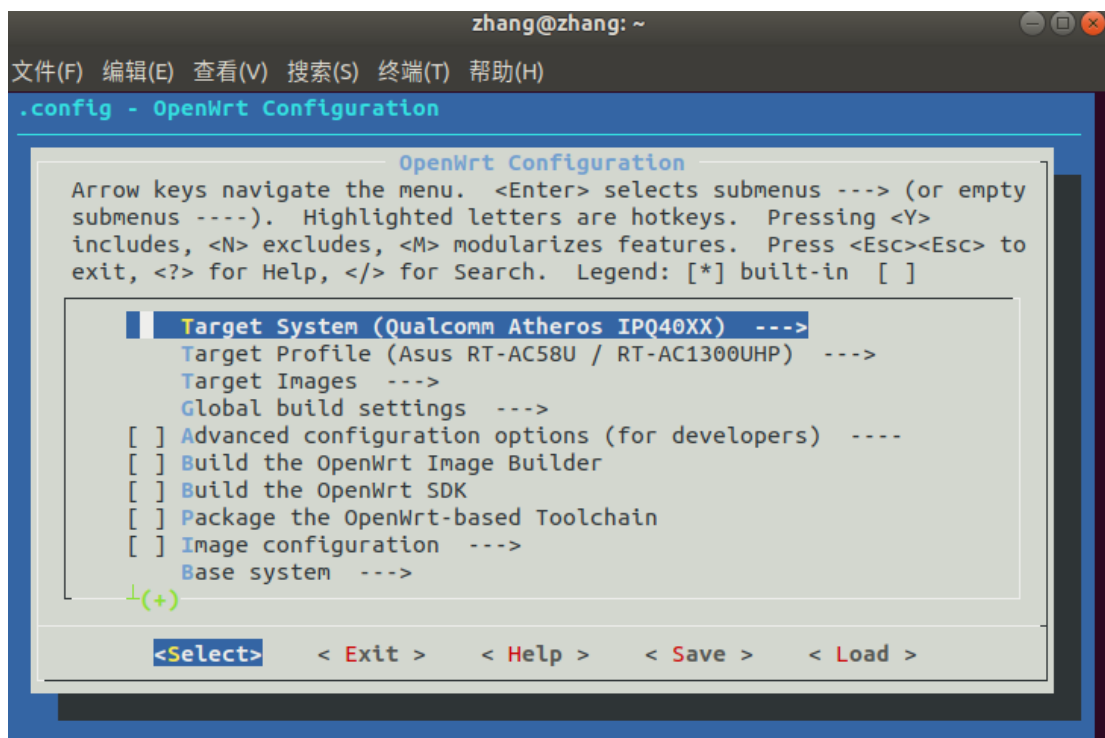
这里我选择 1 给你们看下区别

```
你的配置文件如下：  
  
ac58u  
Adslr_G7  
ea6350  
ea6350_1  
G7_4.14.93  
G7_lean  
Linksys_ea6350  
My_config  
newifi3-usb  
openwrtx86  
test  
tp6300  
wrt32x  
x86  
X86  
x86-test  
  
请输入你要调用的配置名（记得区分大小写）：
```

这是我之前保存的配置，这里选择 **ac58u**

```
正在调用  
配置加载完成  
  
3秒以后执行代码  
如果不需要执行代码以Ctrl+c 终止即可  
  
Collecting package info: package/feeds/packages/golang-torproject-pluggable-tran  
Collecting package info: done  
Collecting target info: target/linux/x86 █
```

配置加载完成，如果不需要改动插件，直接 **exit** 即可



改动过插件选择 1

没改动过插件选择 2

```
-----
是否要保存你的配置，以备下次使用(1.是 2.否 )
注：同一名字的文件会覆盖
-----
请输入你的决定：
```

PS: 配置名字如果一样会直接覆盖，切记

选择 1 编译固件

```
-----
请选择编译固件 OR 编译插件
1.编译固件
2.编译插件
-----
请输入你的决定：
```

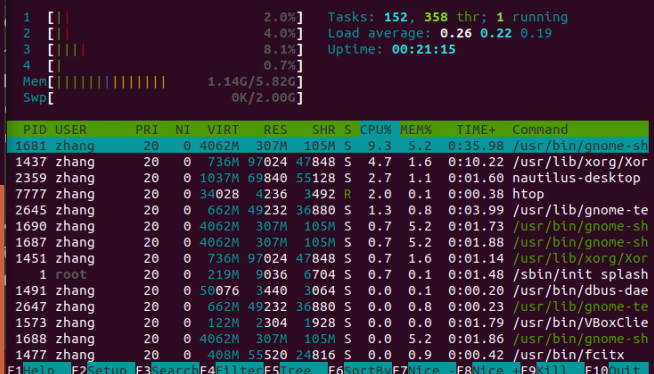
PS: 2 编译插件必须要先编译过固件，不然准报错

编译插件是否用多线程

编译插件是否要使用多线程编译

首次编译不建议，具体用几线程看你电脑，不懂百度，有机会编译失败 回车默认运行make V=s,多线程例子: make -j3 V=s -j (这个值看你电脑)，不要随便乱输，电脑炸了不管，如果你不需要多线程编译那么直接回车即可

请输入你的参数(回车默认: make V=s): make -j3 V=s



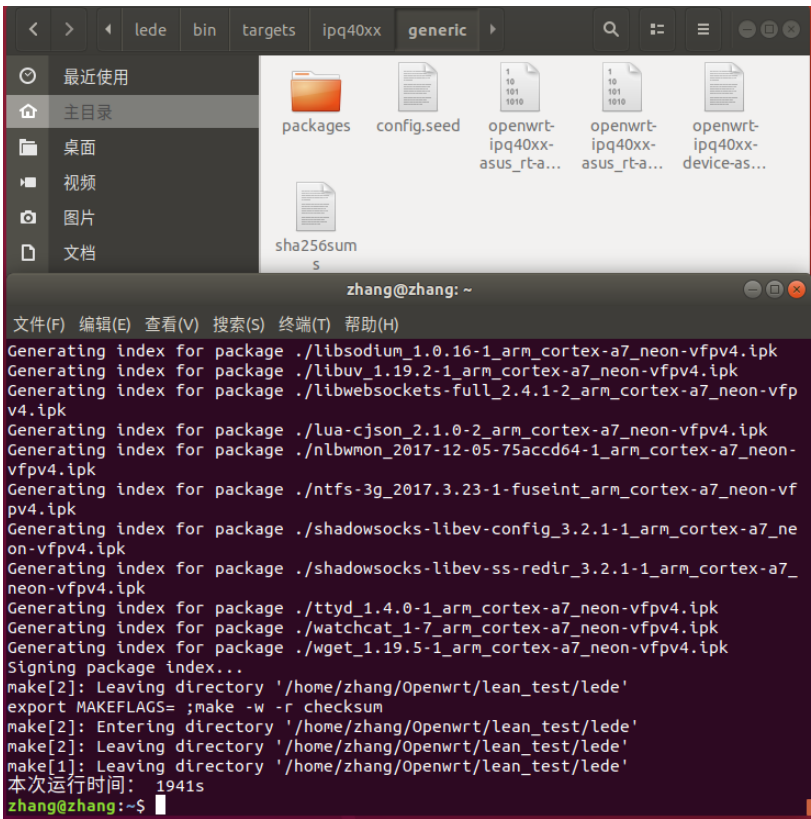
开始漫长的等待了，大概几个小时就行了，二次编译可以-j，-j 会快点，

```
你输入的线程是: make -j3 V=s
准备开始执行编译

3秒以后执行代码
如果不需要执行代码以ctrl+c 终止即可

make[1]: Entering directory '/home/zhang/Openwrt/test/lede'
make[2]: Entering directory '/home/zhang/Openwrt/test/lede'
make[2]: Entering directory '/home/zhang/Openwrt/test/lede'
```

速度还不错，给力



4.5 更新源码

选择一下你的文件夹

```
-----
准备开始更新openwrt源代码与软件
-----
***你的openwrt文件夹有以下几个***
lean_test
test
请选择你要输入你要更新的文件夹: █
```

自动执行 make clean

```
zhang
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
开始清理之前的编译文件
make[1] cclean
```

如果自己修改过源码，直接选择 1,

```
zhang@zhang: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
有没有改动过源代码，因为改动过源代码可能会导致git pull失效无法更新
1.是
2.否
请输入你的决定: █
```

PS: 选择 1 以后，你修改的源码会不见，会以最新的源码为主，建议提前把自己修改的部分 copy 出来，后面再还原回去

执行结果:

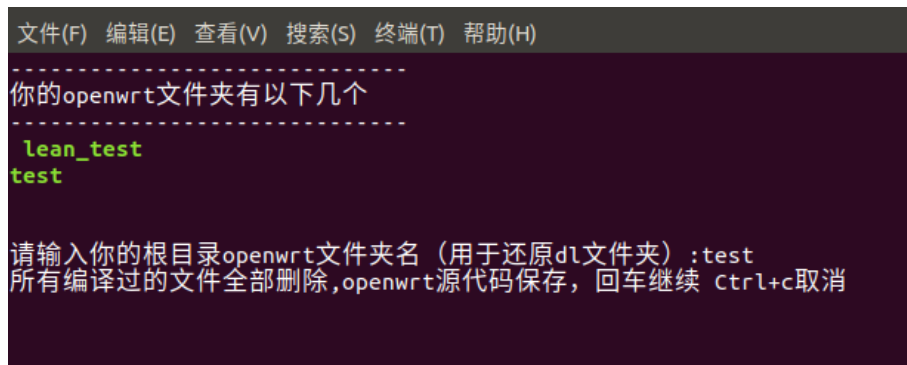
```
请输入你的决定: 1
正在获取 origin
HEAD 现在位于 65b25ceb ramips: add wm8960 asoc sound driver for mt762x

Collecting package info: done
Updating feed 'routing' from 'https://git.openwrt.org/feed/routing.git;openwrt-18.06' ...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 2), reused 0 (delta 0)
展开对象中: 100% (5/5), 完成.
来自 https://git.openwrt.org/feed/routing
6e50f8b..049cb8a openwrt-18.06 -> origin/openwrt-18.06
更新 6e50f8b..049cb8a
Fast-forward
 babeld/Makefile | 5 ++---
 1 file changed, 2 insertions(+), 3 deletions(-)
Create index file './feeds/routing.index'
Collecting package info: done
Installing all packages from feed luci.
Installing all packages from feed packages.
Installing all packages from feed routing.

更新完成回车进行编译  ctrl+c取消,不进行编译
```


4.6 恢复编译环境（危险操作）

主要用于编译环境破坏或者你修改了代码，一直报错，没办法的时候用



回车就可以进行编译了，里面什么都没有了，重新编译又要好长时间了

4.7 Openwrt 官网源码添加 lean 插件

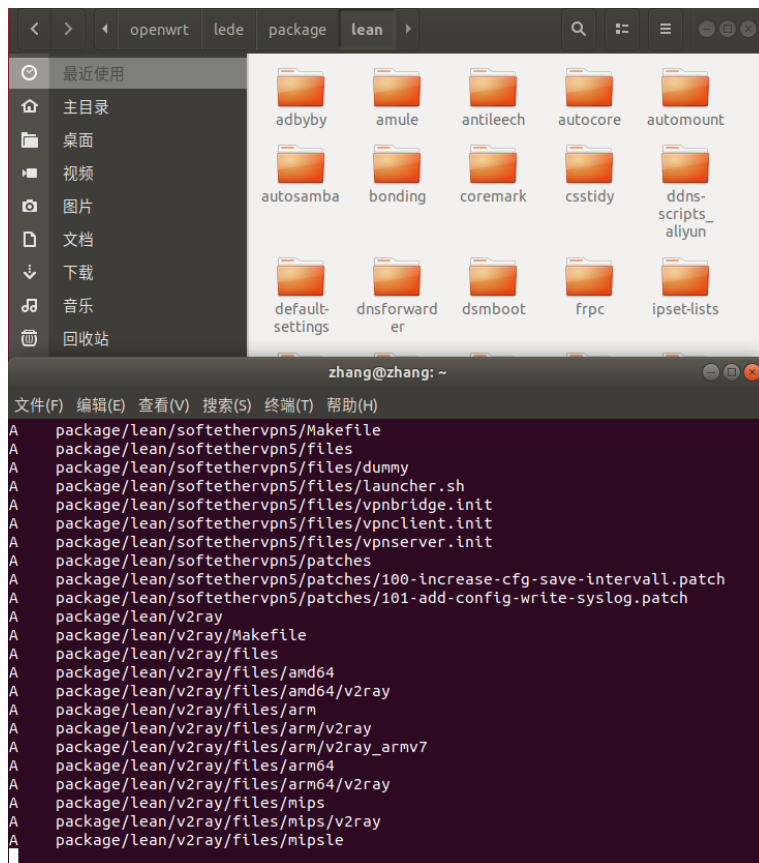
脚本在检查到官方的源码的时候，会自己把 **lean** 的软件库搬过去，所以就会涉及一个更新问题，其他选项等下再说

```
zhang@zhang: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
准备下载openwrt代码  
  
1.Lean_R8(stable version)_source  
2.Lean_R9(Trunk)_source  
3.openwrt17.1(stable version)_source  
4.openwrt18.6(stable version)_source  
5.openwrt19.7(stable version)_source  
6.openwrt(Trunk)_source  
  
0.exit  
  
-----  
请输入你要下载的源代码:6  
正克隆到 'lede'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (3/3), done.  
接收对象中: 0% (1/483143)
```

脚本自动检查

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
  
-----  
检测到你是openwrt官方源码，是否加入lean插件  
1.添加插件(测试功能会有问题)  
2.不添加插件  
-----  
请输入你的选择:█
```

Lean 插件测试完成的只有 **55r**，**frp**，其他插件不怎么用，如果你会修复，麻烦修复完成提交一下，这样大家就可以在新版 **openwrt** 里面用 **lean** 的插件了



更新一下 feeds

```
-----
更新Feeds代码
-----
Updating feed 'luci' from 'https://github.com/coolsnowwolf/luci' ...
正克隆到 './feeds/luci'...
remote: Enumerating objects: 3970, done.
remote: Counting objects: 100% (3970/3970), done.
remote: Compressing objects: 100% (1886/1886), done.
remote: Total 3970 (delta 828), reused 3775 (delta 767), pack-reused 0
接收对象中: 100% (3970/3970), 3.71 MiB | 543.00 KiB/s, 完成.
处理 delta 中: 100% (828/828), 完成.
Create index file './feeds/luci.index'
Checking 'working-make'... ok.
Checking 'case-sensitive-fs'... ok.
Checking 'proper-umask'... ok.
Checking 'gcc'... ok.
Checking 'working-gcc'... ok.
Checking 'g++'... ok.
Checking 'working-g++'... ok.
Checking 'ncurses'... ok.
Checking 'perl-thread-queue'... ok.
Checking 'tar'... ok.
Checking 'find'... ok.
Checking 'bash'... ok.
```

测试编译环境

```
-----
测试编译环境
-----
collecting package info: package/feeds/packages/gnunet
```

下载 dl 选择 2

选择DL服务器

- 1. 国内DL服务器，下载更快
- 2. 官方的DL服务器（需要梯子，不然容易报错）

请输入你的决定： █

开始等待吧

开始下载DL，如果出现下载很慢，请检查你的梯子

3秒以后执行代码
如果不需要执行代码以Ctrl+C 终止即可

```
+ mkdir -p /home/test/Openwrt/test/lede/staging_dir/toolchain-x86_64_gcc-7.4.0_musl
+ cd /home/test/Openwrt/test/lede/staging_dir/toolchain-x86_64_gcc-7.4.0_musl
+ ln -nsf lib lib64
+ ln -nsf lib lib32
+ mkdir -p stamp lib usr/include usr/lib
```

没啥问题就选择 2，有就选择 1，一般有问题都是网络问题

```
make[2]: Nothing to be done for 'download'.
make[2]: Leaving directory '/home/zhang/Openwrt/test/lede/package/utils/lua'
time: package/utils/lua/download#0.10#0.02#0.12
make[2]: Entering directory '/home/zhang/Openwrt/test/lede/package/utils/util-linux'
make[2]: Nothing to be done for 'download'.
make[2]: Leaving directory '/home/zhang/Openwrt/test/lede/package/utils/util-linux'
time: package/utls/util-linux/download#0.10#0.02#0.12
make[1]: Leaving directory '/home/zhang/Openwrt/test/lede'
make[1]: Entering directory '/home/zhang/Openwrt/test/lede'
make[2]: Entering directory '/home/zhang/Openwrt/test/lede/target/linux'
make[3]: Entering directory '/home/zhang/Openwrt/test/lede/target/linux/x86'
make[3]: Nothing to be done for 'download'.
make[3]: Leaving directory '/home/zhang/Openwrt/test/lede/target/linux/x86'
make[2]: Leaving directory '/home/zhang/Openwrt/test/lede/target/linux'
time: target/linux/download#0.03#0.00#0.05
make[1]: Leaving directory '/home/zhang/Openwrt/test/lede'
```

请检查上面有没有error出现，如果有请重新下载

- 1. 有
- 2. 没有

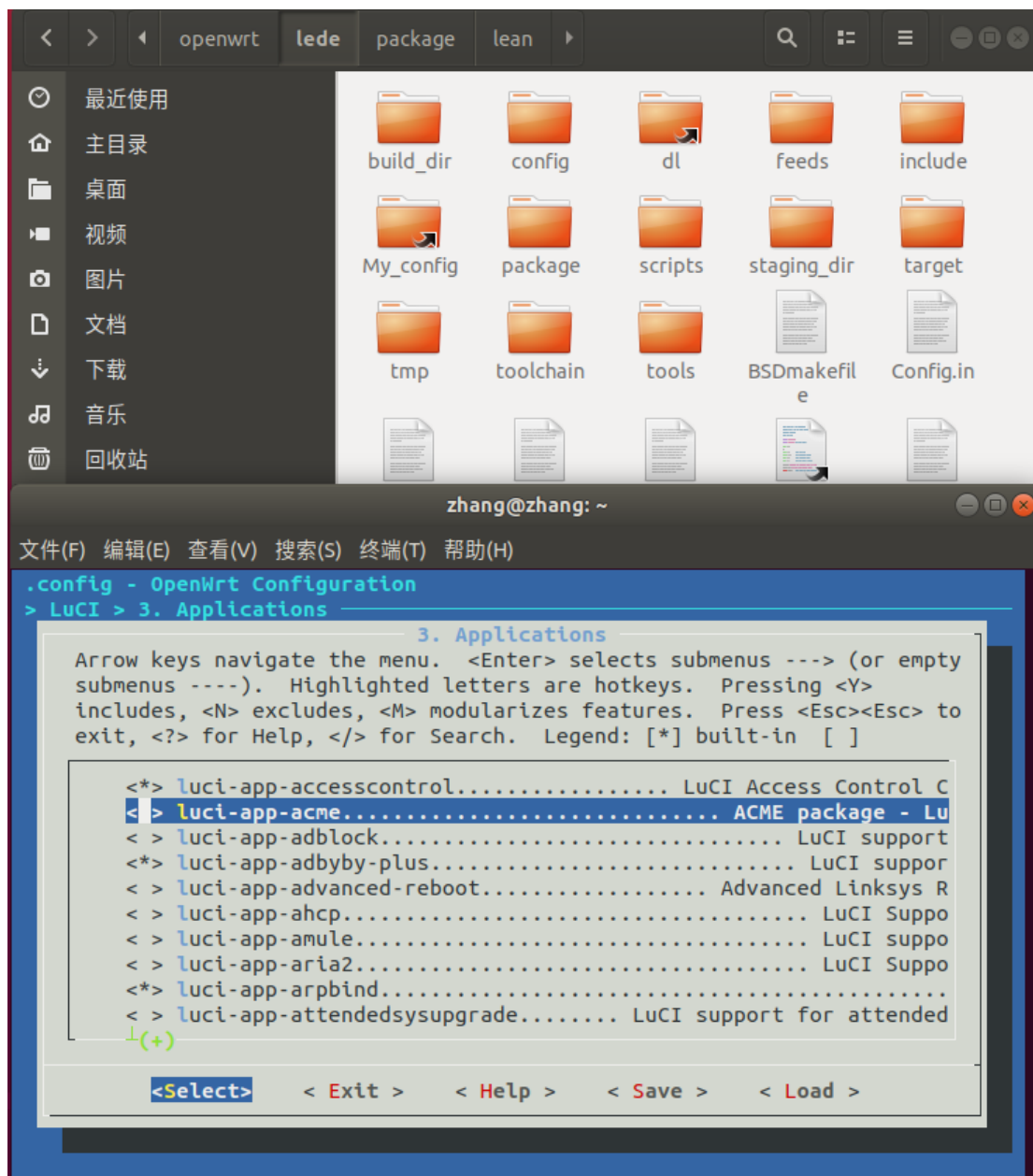
请输入你的决定： █

【××编译环境搭建成功××】

- 1. 请回车进入编译菜单，第一次回车较慢稍等
- 2. 进去编译菜单选择你要的功能完成以后Save
- 3. 菜单Exit以后会自动开始编译

注：如果不需要编译 Ctrl+C退出

搞定，插件加载进来了

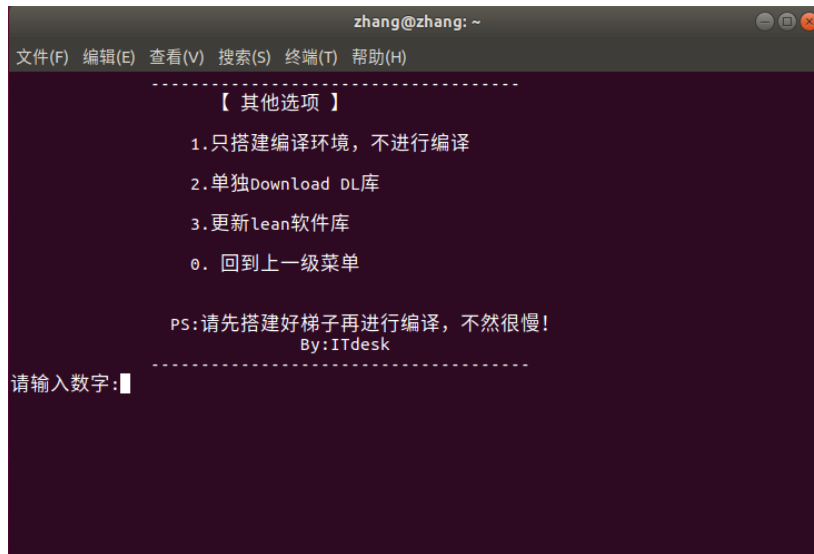


已知 ok 的 55r , frp

其他的欢迎补充，或者提供修改办法

4.8 其他选项

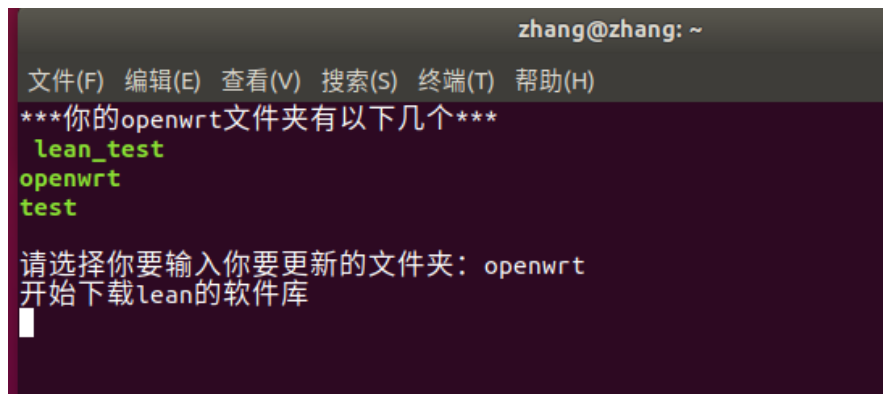
这个选项就是把一下操作单独拿了出来，具体有以下几个



如图所示

1. 只搭建编译环境，不进行编译
2. 单独下载 dl 库
3. 用于更新 openwrt 源码里面 lean 插件

单独说下 3 吧



这个 openwrt 是刚才下载的，用的是官方的源码+lean 插件，现在更新一下插件

更新一下 feeds

```
-----
更新Feeds代码
-----
Updating feed 'luci' from 'https://github.com/coolsnowwolf/luci' ...
正克隆到 './feeds/luci'...
remote: Enumerating objects: 3970, done.
remote: Counting objects: 100% (3970/3970), done.
remote: Compressing objects: 100% (1886/1886), done.
remote: Total 3970 (delta 828), reused 3775 (delta 767), pack-reused 0
接收对象中: 100% (3970/3970), 3.71 MiB | 543.00 KiB/s, 完成.
处理 delta 中: 100% (828/828), 完成.
Create index file './feeds/luci.index'
Checking 'working-make'... ok.
Checking 'case-sensitive-fs'... ok.
Checking 'proper-umask'... ok.
Checking 'gcc'... ok.
Checking 'working-gcc'... ok.
Checking 'g++'... ok.
Checking 'working-g++'... ok.
Checking 'ncurses'... ok.
Checking 'perl-thread-queue'... ok.
Checking 'tar'... ok.
Checking 'find'... ok.
Checking 'bash'... ok.
```

测试编译环境

```
-----
测试编译环境
-----
Collecting package info: package/feeds/packages/gnunet
```

下载 dl 选择 2

```
选择DL服务器
-----
1.国内DL服务器，下载更快
2.官方的DL服务器（需要梯子，不然容易报错）
-----
请输入你的决定：2
```

开始等待吧

```
-----
# 开始下载DL，如果出现下载很慢，请检查你的梯子 #
-----

3秒以后执行代码
如果不需执行代码以ctrl+c 终止即可

+ mkdir -p /home/test/Openwrt/test/lede/staging_dir/toolchain-x86_64_gcc-7.4.0_musl
+ cd /home/test/Openwrt/test/lede/staging_dir/toolchain-x86_64_gcc-7.4.0_musl
+ ln -nsf lib lib64
+ ln -nsf lib lib32
+ mkdir -p stamp lib usr/include usr/lib
█
```

没啥问题就选择 2，有就选择 1，一般有问题都是网络问题

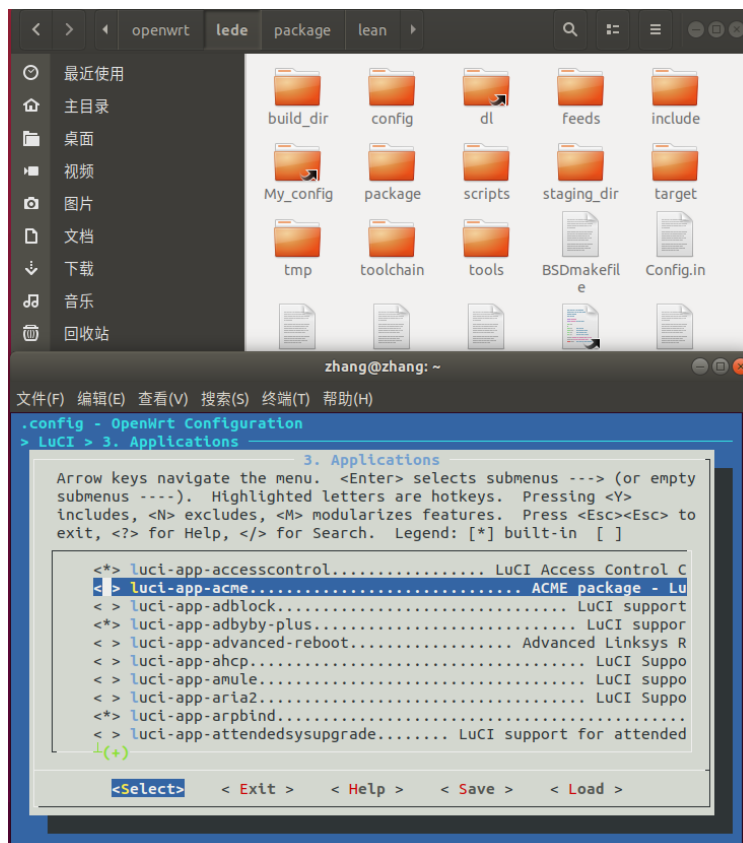
```
make[2]: Nothing to be done for 'download'.
make[2]: Leaving directory '/home/zhang/Openwrt/test/lede/package/utls/lua'
time: package/utls/lua/download#0.10#0.02#0.12
make[2]: Entering directory '/home/zhang/Openwrt/test/lede/package/utls/util-linux'
make[2]: Nothing to be done for 'download'.
make[2]: Leaving directory '/home/zhang/Openwrt/test/lede/package/utls/util-linux'
time: package/utls/util-linux/download#0.10#0.02#0.12
make[1]: Leaving directory '/home/zhang/Openwrt/test/lede'
make[1]: Entering directory '/home/zhang/Openwrt/test/lede'
make[2]: Entering directory '/home/zhang/Openwrt/test/lede/target/linux'
make[3]: Entering directory '/home/zhang/Openwrt/test/lede/target/linux/x86'
make[3]: Nothing to be done for 'download'.
make[3]: Leaving directory '/home/zhang/Openwrt/test/lede/target/linux/x86'
make[2]: Leaving directory '/home/zhang/Openwrt/test/lede/target/linux'
time: target/linux/download#0.03#0.00#0.05
make[1]: Leaving directory '/home/zhang/Openwrt/test/lede'
-----
请检查上面有没有error出现，如果有请重新下载
1.有
2.没有
-----
请输入你的决定: 2
```

```
-----
【**编译环境搭建成功**】

1.请回车进入编译菜单，第一次回车较慢稍等
2.进去编译菜单选择你要的功能完成以后save
3.菜单Exit以后会自动开始编译

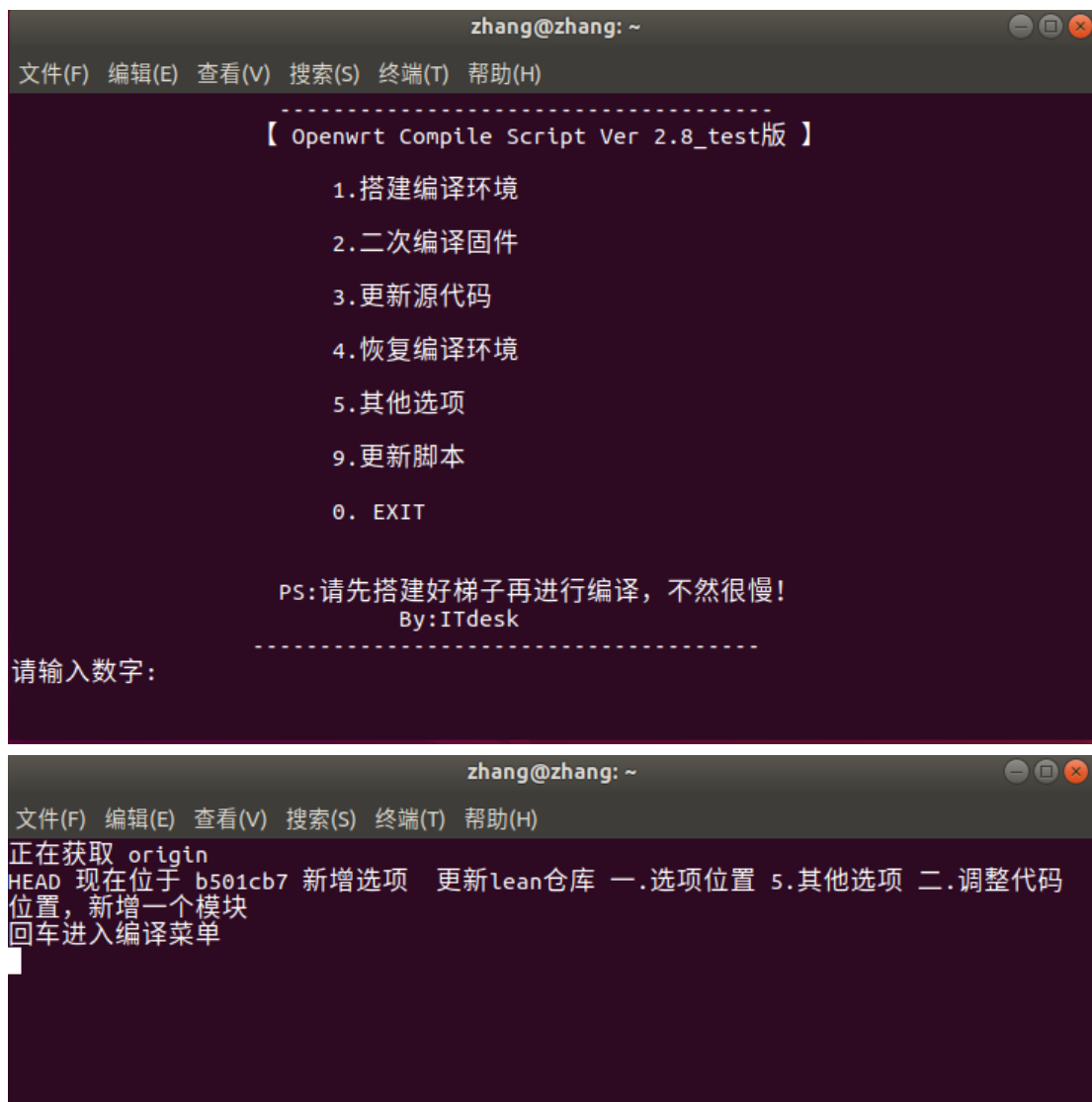
注：如果不需要编译 Ctrl+c退出
-----
```

搞定，插件加载进来了（更新完成）



4.9 更新脚本

顾名思义



The image shows two screenshots of a terminal window. The top screenshot displays the main menu of the 'Openwrt Compile Script Ver 2.8_test版'. The menu options are: 1.搭建编译环境, 2.二次编译固件, 3.更新源代码, 4.恢复编译环境, 5.其他选项, 9.更新脚本, and 0.EXIT. A note at the bottom says 'ps:请先搭建好梯子再进行编译, 不然很慢!' and 'By:ITdesk'. The prompt '请输入数字:' is shown. The bottom screenshot shows the script's execution progress: '正在获取 origin HEAD 现在位于 b501cb7 新增选项 更新lean仓库 一.选项位置 5.其他选项 二.调整代码位置, 新增一个模块 回车进入编译菜单'.

```
zhang@zhang: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
-----  
【 Openwrt Compile Script Ver 2.8_test版 】  
  
1.搭建编译环境  
2.二次编译固件  
3.更新源代码  
4.恢复编译环境  
5.其他选项  
9.更新脚本  
0.EXIT  
  
ps:请先搭建好梯子再进行编译, 不然很慢!  
By:ITdesk  
-----  
请输入数字:
```

```
zhang@zhang: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
正在获取 origin  
HEAD 现在位于 b501cb7 新增选项 更新lean仓库 一.选项位置 5.其他选项 二.调整代码  
位置, 新增一个模块  
回车进入编译菜单
```

回车就行了

使用教程编写完成

By: ITdesk

版本: 1.0

2019.09.27