

下载：

```
git clone ssh://luther.ge@10.5.41.60:29418/toolchain/clion
```

└─> 改成自己的名字

一. 激活与配置

1. 激活

a. 执行 install.sh

首先进入下载下来的 clion 所在目录，执行 install.sh 这个脚本：

```
SPREADTRUM\qiaoting.lin@tj08294pcu:~/vobs/tools/clion$ ./install.sh
```

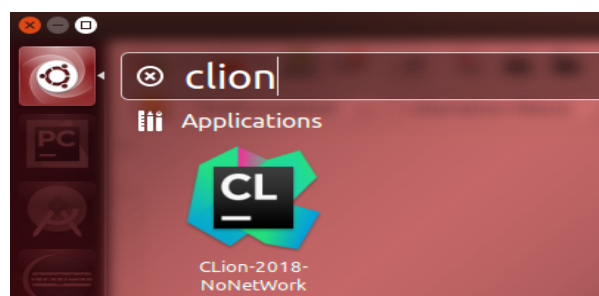
```
SPREADTRUM\qiaoting.lin@tj08294pcu:~/vobs/tools/clion$ ls
bin          clion-2018.settings.ExportSettings-20190529.jar  clion使用文档.odt      help          JetBrainsCrack-2.8-release-enc.jar  lib      README.odt
build.txt    clion.desktop                                     clion按键手法_20190418_074735.jpg  Install-Linux-tar.txt  jre64      license  unshare
check.sh     clion.sh                                           _cl_.sh                  install.sh      key.txt    plugins
```

b. 打开应用

点击桌面左上角

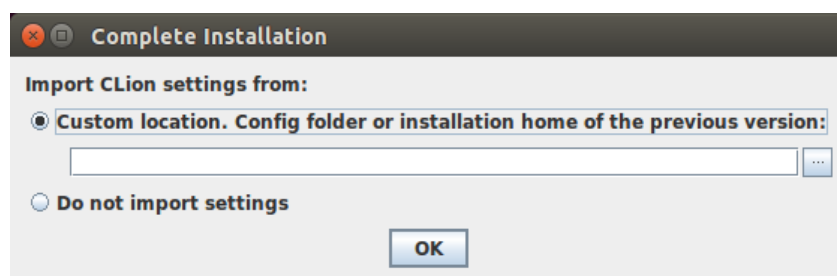


，搜索 clion, 并打开应用



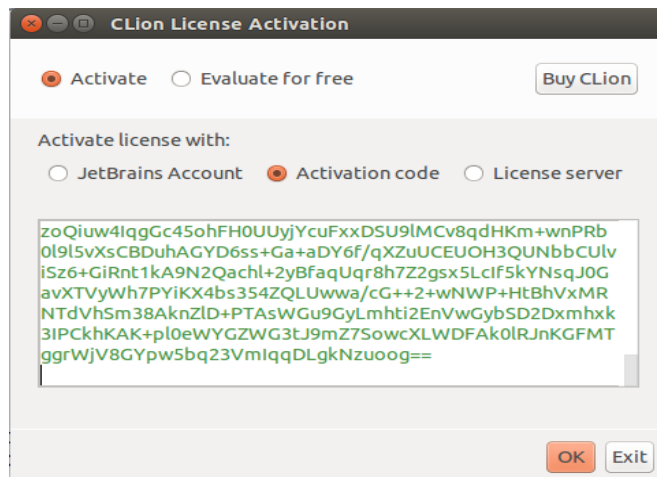
c. 根据提示按步骤激活

打开 clion 之后，弹出如下对话框： 选择 "Do not import settings"， 点击 ok



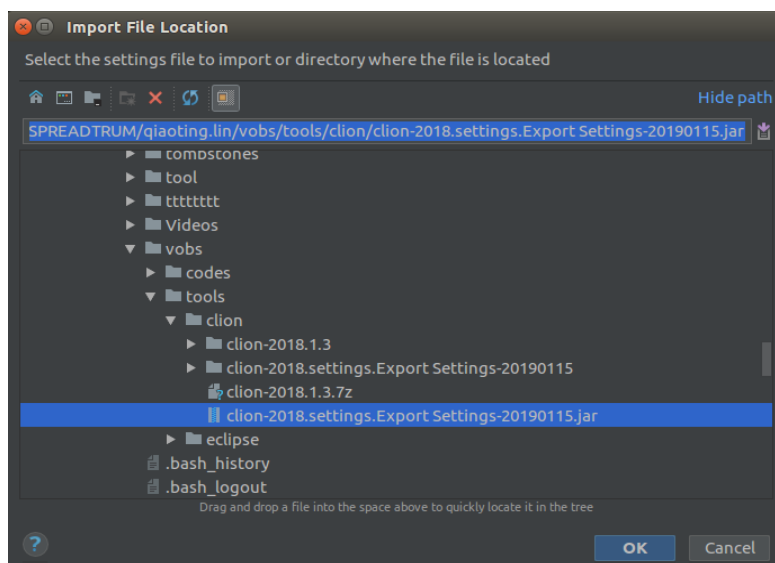
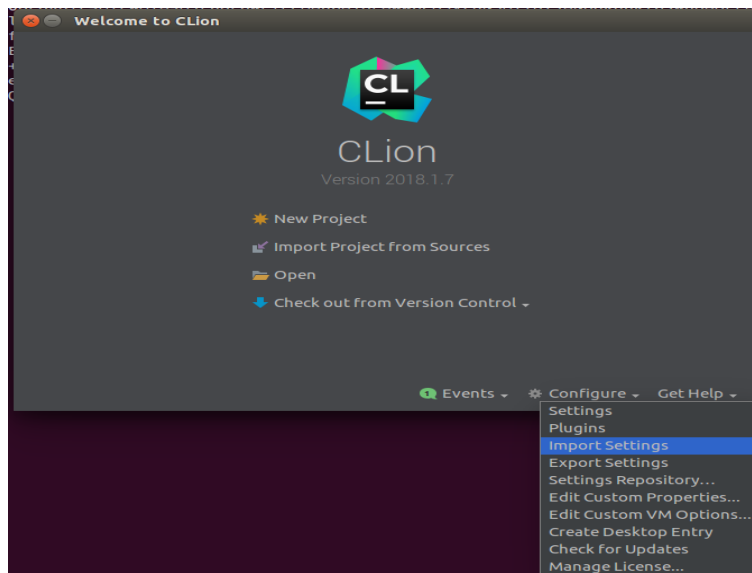
接下来会进入激活页面，如下图所示，完成激活。

选择 “ Activate → Activation code → 将 clion 目录下 key.txt 中的激活码拷贝到文本框 → OK ”



2. 配置

激活完后会弹出应用导航界面，选择 open 打开应用，
单击菜单栏 file → Import Settings → 选择 clion 目录下配置文件，即 clion-
2018.settings.ExportSettings-20190529.jar → OK



二. Clion 工程的建立（快捷方式）

快捷方式就是直接拷贝 luther 建好的工程 ^_^

1. 拷贝 clion 工程：

进入我们 tool 仓库，先执行 git pull 更新一下仓库

(tool 仓库: `git clone git://10.5.41.60/tool`)

然后将/luther/clion工程拷贝到源码的 art 仓库下：

`cp -rf clion ~/sprdroid9.0_trunk/art/`

```
SPREADTRUM\qiaoting.lin@tj08294pcu:~/tool/luther$ ls
clion  README  run-sp9832e_1h10_oversea-user-tiny@zebu, native, antutu7apk.no3D.sh  run.u-boot.S.sh  sprdroid8.1_trunk-hidl.sources.7z  tiny-android9.0
SPREADTRUM\qiaoting.lin@tj08294pcu:~/tool/luther$ cp -rf clion ~/sprdroid9.0_trunk/art/
```

2. 修改 CMakeLists.txt

进入源码~/sprdroid9.0_trunk/art/clion 目录, 打开 CMakeLists.txt

修改第 3 行的 ANDROID_ROOT 变量：改成自己的 Android 源码路径

```
TRUM\qiaoting.lin@tj08294pcu: ~/sprdroid9.0_trunk/art/clion
SPREADTRUM\qiaoting.lin@tj08294pcu: ~/sprdroid9.0_trunk... x  SP
cmake_minimum_required(VERSION 3.0)
set(PROJECT_NAME "native-all-android9.0")
set(ANDROID_ROOT "~/sprdroid9.0_trunk")
set(CLION_ROOT "out/development/ide/clion")
project(${PROJECT_NAME})
```

三. 新建 Clion 工程

由于前面快捷方式所建的工程非常大，我们可以自己新建一个工程，其中只包含自己模块涉及的代码

1. 生成 CMakeLists.txt

a. 建一个工程文件夹，如 sprdroid9.0_trunk_code，然后将你关心的代码目录以软链接的形式链接到你本地源码所在目录，这里软链接了如下几个目录的文件：

包括 art, bionic, libcore, external/selinux 以及 system/core,

可以看到，它们都软链接到了我本地源码 (sprdroid9.0_trunk) 对应的文件了。

```
SPREADTRUM\qiaoting.lin@tj08294pcu:/home/newdisk/clion_test/sprdroid9.0_trunk_code$ ls -l
total 8
lrwxrwxrwx 1 SPREADTRUM\qiaoting.lin SPREADTRUM\domain^users 35 6月 1 16:25 art -> /home/newdisk/sprdroid9.0_trunk/art
lrwxrwxrwx 1 SPREADTRUM\qiaoting.lin SPREADTRUM\domain^users 38 6月 1 16:20 bionic -> /home/newdisk/sprdroid9.0_trunk/bionic
drwxr-xr-x 2 SPREADTRUM\qiaoting.lin SPREADTRUM\domain^users 4096 6月 1 16:24 external
lrwxrwxrwx 1 SPREADTRUM\qiaoting.lin SPREADTRUM\domain^users 39 6月 1 16:23 libcore -> /home/newdisk/sprdroid9.0_trunk/libcore
drwxr-xr-x 2 SPREADTRUM\qiaoting.lin SPREADTRUM\domain^users 4096 6月 1 16:21 system
SPREADTRUM\qiaoting.lin@tj08294pcu:/home/newdisk/clion_test/sprdroid9.0_trunk_code$ ls -l external/
total 0
lrwxrwxrwx 1 SPREADTRUM\qiaoting.lin SPREADTRUM\domain^users 48 6月 1 16:24 selinux -> /home/newdisk/sprdroid9.0_trunk/external/selinux
SPREADTRUM\qiaoting.lin@tj08294pcu:/home/newdisk/clion_test/sprdroid9.0_trunk_code$ ls -l system/
total 0
lrwxrwxrwx 1 SPREADTRUM\qiaoting.lin SPREADTRUM\domain^users 43 6月 1 16:19 core -> /home/newdisk/sprdroid9.0_trunk/system/core
```

b. 执行 bcmake-files

在当前目录 sprdroid9.0_trunk_code 下，执行本地源码/vendor/sprd/proprieties-source/tiny-formatter/tool 目录下的 bcmake-files，如下所示：

```
SPREADTRUM\qiaoting.lin@tj08294pcu:/home/newdisk/clion_test/sprdroid9.0_trunk_code$ /home/newdisk/sprdroid9.0_trunk/vendor/sprd/proprieties-source/tiny-formatter/tool/bcmake-files
```

➡ 这是我本地的 Android9.0 代码

注意:是在你当前新建的这个工程，即 `sprdroid9.0_trunk_code` 目录下，执行你本地代码 `vendor` 目录下的 `bcmake-files`。

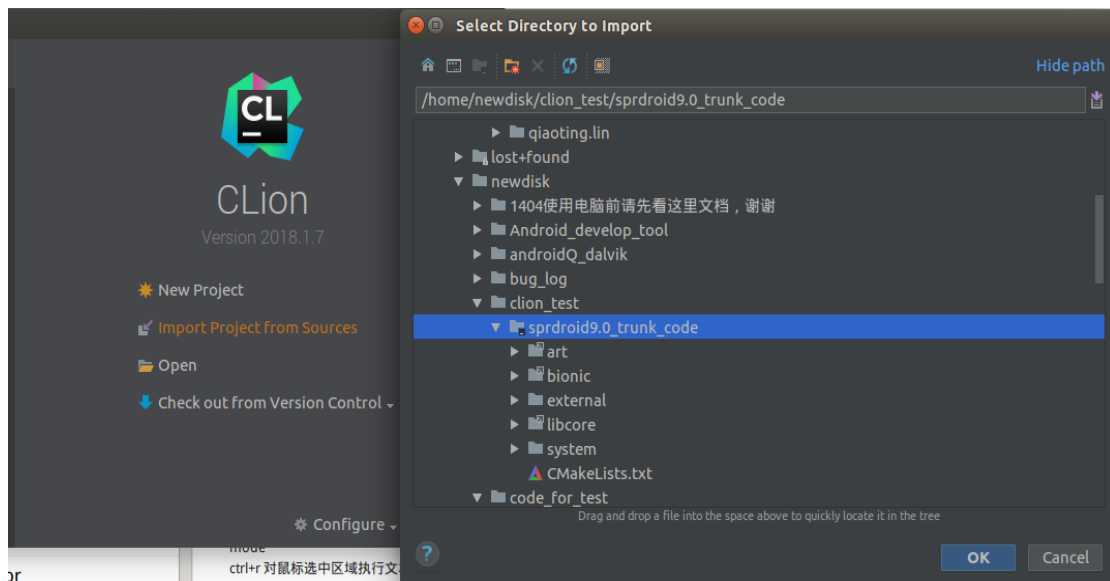
执行完后，可以看到当前目录下生成了 `CMakeLists.txt`

```
SPREADTRUM\qiaoting.lin@tj08294pcu:/home/newdisk/clion_test/sprdroid9.0_trunk_code$ /home/newdisk/sprdroid9.0_trunk/vendor/sprd/proprieties-source/tiny-formatter/tool/bcmake-files
SPREADTRUM\qiaoting.lin@tj08294pcu:/home/newdisk/clion_test/sprdroid9.0_trunk_code$ ls
art  bionic  CMakeLists.txt  external  libcore  system
```

2. 打开工程

最后，打开 `clion`，选择 `Import Project from Sources`，选择我们新建的这个工程，即 `sprdroid9.0_trunk_code`，点击“OK”即可。

工程导入会耗时几分钟，耐心等待...^_^



四.Clion 常用快捷键

主要用到的快捷键如下 2 个表。

表一

快捷键	说明
光标选中， 然后 Alt+鼠标滚轮	可以在多个出现的地方滚动跳转
ctrl+8	Show usage 函数或变量使用信息快速提示框
ctrl+]	Find usage 函数或变量使用信息详细展示，使用左下角窗体的上下箭头快速在代码文件遍历查看
ctrl+0	全局查找字符串
ctrl+n	查找类定义
ctrl+\	查找 symbol 定义
ctrl+p	往前跳
ctrl+[往后跳
ctrl+i	调用关系层次结构 Call Hierarchy
ctrl+h	列出类的继承关系，比如一个 virtual 方法，可以通过此方式找到类然后，进一步找到继承类对应的 virtual 方法实现
ctrl+-	折叠光标所在函数或类
ctrl+=	展开光标所在函数或类
ctrl+shift+n	打开文件
ctrl+shift+alt+n	查找 symbol 定义
ctrl+shift+[跳转到当前代码所在函数或者类头部
ctrl+shift+]	跳转到当前代码所在函数或者类尾部
ctrl+shift+m	在靠近的最外层 caret 之间跳转，即括号匹配体中来回跳转
ctrl+shift+Backtrace	回到上一次编译的地方 Last Edit Location
ctrl+shift+alt+Backtrace	回到下一次编辑的地方（自定义快捷键）Next Edit Location
ctrl+left 或 right	前后按单词跳转
ctrl+g	跳转到指定行
ctrl+e	最近打开过的文件
alt+shift+c	查看最近修改过的文件（快速定位最近修改的文件挺方便的）
ctrl+q	显示注释文档
alt+↑/↓	快速定位方法头部

表二： 常用的编辑命令

快捷键	说明
ctrl+z	取消
ctrl+shift+z	撤销取消
alt+shift+Insert	多列选择模式,可以多列同时删除或者插入 Column selection mode
ctrl+r	对鼠标选中区域执行文本替换或者全文文本替换（像 vim 鼠标选中的 shift + : s/xx/gg/g，很方便）
ctrl+/ ctrl+shift+/ ctrl+shift+j	注释本行// 多行/**/注释 拼接 2 行到 1 行
ctrl+shift+u	大小写转化
alt+/ ctrl+w	单词补齐（像 vim 的 ctrl+n，很方便） 选中代码，连续按会扩大范围，自动选中更大范围代码
ctrl+上或下方向键	整屏向上或者向下滚动一行（类似 vim 的 ctrl+e 和 ctrl+y）
ctrl+x	删除行
ctrl+d	复制行