

Clang编译环境配置

1. 安装 Linux 子系统，CMD内运行

```
lxrun /install
```

通过商店安装的暂时没找到在哪里，用CMD安装好的Linux系统目录在：[C:\Users\用户名\AppData\Local\lxss](#)

如果之前有安装Linux子系统先卸载，再安装

```
lxrun /uninstall /full
```

安装完之后在CMD内输入**bash**就可以打开Linux子系统

2. 新建 root 帐户

```
sudo passwd root
```

3. 系统常规更新

```
su  
apt-get update  
apt-get upgrade
```

4. 安装必要的包

```
# svn  
apt install subversion  
# cmake  
apt install cmake  
# build-essential  
apt-get install build-essential
```

5. 先解压CL压缩包内的CL文件夹到root/目录下，然后

```
cd ~  
cd CL  
ls
```

```
# 若能正常显示build、llvm两个文件夹则为正常
```

6. **非必要（先跳到第7步，若第7步可用就不用执行）

```
# 先删除build文件夹，因为build文件夹是在我本机创建的，其他机器不一定能用
rm -rf build
# 重新创建build文件夹
mkdir build
cd build
# 用gcc和g++编译Clang的源码，得到Clang的包
cmake ../llvm -DLLVM_TARGETS_TO_BUILD=X86 -DCMAKE_BUILD_TYPE=Release
make -j4
# 安装Clang
make install
# 判断Clang是否安装成功，正常显示版本号就是正常的
clang++ --version
```

7. 安装Clang编译器

```
# 安装Clang
cd build
make install
# 判断Clang是否安装成功，正常显示版本号就是正常的
clang++ --version
```

8. 测试Clang编译

```
cd ~
cd CL
# 使用Clang编译Clang的源码
mkdir test1
cd test1
CC=clang CXX=clang++ cmake ../llvm -DLLVM_TARGETS_TO_BUILD=X86 -
DCMAKE_BUILD_TYPE=Release
make -j4
```

可以重复测试，切换到CL目录新建一个test文件夹就行了

```
cd ~
cd CL
# 使用Clang编译Clang的源码
mkdir test2
```

```
cd test2
CC=clang CXX=clang++ cmake ../llvm -DLLVM_TARGETS_TO_BUILD=X86 -
DCMAKE_BUILD_TYPE=Release
make -j4
```