

# Rid 安装配置笔记

---

## 1、安装环境

- VMWare Workstation 12.0 虚拟机
- 2核4线程, 8G内存
- Ubuntu-14.04.5-desktop-amd64

## 2、安装依赖软件

- 在命令行下输入：

```
~$ sudo apt-get install cmake python-igraph libxml2-dev libncurses5-dev git  
vim g++ build-essential libboost-all-dev  
~$ sudo apt-get update
```

- 从github上下载llvm, 然后进入llvm目录, 用git checkout命令切换到rid所使用的版本

```
~$ git clone https://github.com/llvm-mirror/llvm.git  
~$ cd llvm/  
~/llvm$ git checkout a383742
```

- 进入llvm/tools/目录, 下载clang, 然后进入clang目录, 用git checkout命令切换到rid所使用的版本。

```
~$ cd llvm/tools/  
~/llvm/tools$ git clone https://github.com/llvm-mirror/clang.git  
~/llvm/tools$ cd clang/  
~/llvm/tools/clang$ git checkout dd1c90c
```

- 编译llvm与clang。在cmake指令后面有一些推荐的参数可以选配, 用来加快编译。

```
~$ cd llvm  
~/llvm$ mkdir build-cmake
```

```
~/llvm$ cd build-cmake
~/llvm/build-cmake$ cmake -DCMAKE_BUILD_TYPE=Release -
DLLVM_ENABLE_RTTI=ON ../
~/llvm/build-cmake$ make -j8
~/llvm/build-cmake$ sudo make install
```

备注：倒数第二句的make -j8命令中，-j8是开启多线程编译的选项，用来加快编译，8是线程个数，要根据实际机器情况进行调整，一般取实际硬件线程数的1到2倍。

- 通过llvm-cov --version与clang --version检查llvm与clang的安装情况，看到version 3.6.0就说明安装成功了。

```
~$ llvm-cov --version
LLVM version 3.6.0svn
...
~$ clang --version
clang version 3.6.0
...
```

- 下载并安装Microsoft Z3（详见z3主页：<https://github.com/Z3Prover/z3>）

```
~$ git clone https://github.com/Z3Prover/z3.git
~$ cd z3
~/z3$ python scripts/mk_make.py
~/z3$ cd build
~/z3/build$ make
~/z3/build$ make install
```

---

### 3、安装Rid

- 编译安装Rid

```
~$ cd rid/src
~/rid/src$ mkdir build
```

```
~/rid/src$ cd build
```

```
~/rid/src/build$ cmake -DCMAKE_BUILD_TYPE=Release ..
```

```
~/rid/src/build$ make install
```

- 测试安装是否成功

```
~$ cd rid/tests
```

```
~/rid/tests$ ./run.sh
```

- 得到如下图的结果，就认为安装成功了（暂时不用管那6个FAIL，等原作者处理）

```
lt@lt-vm:~/rid/tests$ ./run.sh
assignment      analysis: [ PASS ]
                 cache 0: [ PASS ]
                 cache I/O: [ PASS ]
basic           analysis: [ PASS ]
                 cache 0: [ PASS ]
                 cache I/O: [ PASS ]
bug             analysis: [ FAIL ]
cast           analysis: [ PASS ]
                 cache 0: [ PASS ]
                 cache I/O: [ PASS ]
containerof     analysis: [ FAIL ]
dpm             analysis: [ FAIL ]
errptr         analysis: [ FAIL ]
global         analysis: [ PASS ]
                 cache 0: [ PASS ]
                 cache I/O: [ PASS ]
if             analysis: [ PASS ]
                 cache 0: [ PASS ]
                 cache I/O: [ PASS ]
int            analysis: [ PASS ]
                 cache 0: [ PASS ]
                 cache I/O: [ PASS ]
py             analysis: [ FAIL ]
ret2           analysis: [ PASS ]
                 cache 0: [ PASS ]
                 cache I/O: [ PASS ]
ret            analysis: [ FAIL ]
lt@lt-vm:~/rid/tests$
```

---

## 4、分析Kernel

- 下载经过原作者修改的linux-3.17.0，修改后可以用clang和llvm进行编译和分析。由于原镜像失效，现在暂时用本地保存的文件，后面再讨论应该在哪里放一份镜像。