安装GCC 8.0.1

mkdir gcc-build-8.0.1

安装gmp

```
tar -Jxvf gmp-6.1.0.tar.xz

mkdir temp #在与gmp-6.1.0同级别的目录下建一个新文件夹，叫temp（自己命名）

cd temp

../gmp-6.1.0/configure --prefix=/usr/local/gmp-6.1.0  --#安装路径是/usr/local/，名字叫gmp-6.1.0）
make
sudo make install
```

安装mpfr

```
tar -xvf mpfr-3.1.4.tar.bz2

cd temp

#先清空temp内的文件，这样安装两个包互补影响，当然也可以另外新建一个文件夹，在你新的文件夹下：

../mpfr-3.1.4/configure --prefix=/usr/local/mpfr-3.1.3 --with-gmp=/usr/local/gmp-8.0.1

make

sudo make install
```

安装mpc

```
tar -xzvf mpc-1.0.3.tar.gz

cd temp

#同样先清空temp文件夹

../mpc-1.0.3/configure --prefix=/usr/local/mpc-1.0.3 --with-gmp=/usr/local/gmp-8.0.1 --with-mpfr=/usr/local/mpfr-3.1.4

make

sudo make install
```

安装gcc

```
tar -xzvf gcc-8.0.1.tar.gz

#再次清空temp首先配置GCC，在temp文件夹中执行以下命令，注意，此时temp和gcc-8.0.*在同级目录下

../gcc-8.0.1/configure --prefix=/usr/local/gcc-8.0 --enable-threads=posix --disable-checking --disable-multilib --enable-languages=c,c++ --with-gmp=/usr/local/gmp-6.1.0 --with-mpfr=/usr/local/mpfr-3.1.4 --with-mpc=/usr/local/mpc-1.0.3

make

sudo make install
```

配置环境变量

```
sudo ln -s /usr/local/gcc-8.0.1/bin/gcc /usr/local/bin/gcc8

sudo ln -s /usr/local/gcc-8.0.1/bin/g++ /usr/local/bin/g++8
```

以上可以先不用管试，直接进行下一步，如果出错，再考虑是否更换gcc

安装要测试的Linux内核

```
unzip linux-master.zip
cd linux-master
```

gcc进行编译 编译内核

```
make CC="usr/local/bin/gcc" defconfig
make CC="usr/local/bin/gcc" kvmconfig
```

修改.config文件

```
CONFIG_KCOV=y

CONFIG_DEBUG_INFO=y

CONFIG_KASAN=y
CONFIG_KASAN_INLINE=y

CONFIG_CONFIGFS_FS=y
CONFIG_SECURITYFS=y
```

保存配置

```
make CC="$GCC/bin/gcc" olddefconfig
```

编译成功后，会有两个文件 bzImage 以及 vmlinux 分别位于

linux-master/vmlinux
linux-master/arch/x86/boot/bzImage

安装Image镜像

```
sudo apt-get install debootstrap
```

建立img目录，将create-image.sh拷贝到目录下

```
mkdir image
chmod +x create-image.sh
./create-image.sh -s 1024
```

最后会得到 ./stretch.img 镜像文件

安装qemu虚拟机

sudo apt-get install qemu-system-x86

qemu-system-x86_64 -m 1G -enable-kvm -drive file=<stretch.img路径>,format=raw -kernel <linux-master路径>/arch/x86/boot/bzImage -append root=/dev/sda

安装go环境

```
tar -xf go1.14.2.linux-amd64.tar.gz
mv go goroot
mkdir gopath
export GOPATH=`pwd`/gopath
export GOROOT=`pwd`/goroot
export PATH=$GOPATH/bin:$PATH
export PATH=$GOROOT/bin:$PATH
```

安装syzkaller

解压syzkaller安装包到 /gopath/src/github.com/，执行make安装

cd gopath/src/github.com/google/syzkaller/
make

安装成功后，会有/bin/syz-manger文件

返回syzkaller文件目录下，建立 my.cfg 文件

```
{
        "target": "linux/amd64",
        "http": "127.0.0.1:56741",
        "workdir": "$GOPATH/src/github.com/google/syzkaller/workdir",
        "kernel_obj": "linux-mater文件夹目录",
        "image": "image文件夹目录/stretch.img",
        "sshkey": "image文件夹目录/stretch.id_rsa",
        "syzkaller": "$GOPATH/src/github.com/google/syzkaller",
        "procs": 8,
        "type": "qemu",
        "vm": {
                "count": 4,
                "kernel": "linux-master文件夹目录/arch/x86/boot/bzImage",
                "cpu": 2,
                "mem": 2048
        }
}
```