

RetDec

1. 搭建retdec运行环境

<https://github.com/avast/retdec>

1. 从官网下载release版本

```
wget https://github.com/avast/retdec/releases/download/v5.0/RetDec-v5.0-Linux-Release.tar.xz
```

2. 解压

```
tar -xvf RetDec-v5.0-Linux-Release.tar.xz -C /your/path/to/install
```

3. 创建docker容器

```
docker run -it --privileged \
-v /work/stu/wtxiao/WorkSpace/retdec_22.04/dockerimages:/work/stu/wtxiao/WorkSpace/retdec_22.04/dockerimages \
--name retdec-x86_64-ubuntu_22.04 ubuntu:22.04 /bin/bash
```

4. 使用，如果还有链接库问题，在本地安装相应链接库：

```
/your/path/to/install/bin/retdec-decompiler xxx.exe
```

2. 工程文件夹说明

假设你在本项目的retdec目录夹下，那么（由于二进制程序github规定存放大小，SPEC以及RetDec-v5.0不在仓库中展示）：

```
.
├── \ retdec
│   ├── result          #存放所有被retdec decompile的二进制程序和结果
│   │   ├── MyTest      #本人的测试集，.o结尾的为原先的二进制程序
│   │   └── SPEC        #存放所有成功通过retdec decompile的SPEC测试集
│   └── RetDec-v5.0
│       ├── bin          #其中存放retdec-decompiler的decompile程序
│       ├── CHANGELOG.md
│       ├── include
│       ├── lib
│       ├── LICENSE
│       ├── LICENSE-PELIB
│       ├── LICENSE-THIRD-PARTY
│       ├── README.md
│       ├── SECURITY.md
│       └── share
```

3.测试总结

- 测试环境：

测试的内容	decompile通过情况	备注	retdec版本
test_case	全部通过		5
(SPEC_X86_64) blender_r_base	通过		5
(SPEC_X86_64) bwaves_r_base	通过		5
(SPEC_X86_64) cactuBSSN_s_base	通过		5
(SPEC_X86_64) cactusBSSN_r_base	通过		5
(SPEC_X86_64) cam4_r_base	不通过	生成的ll过大, 存在bug	5
(SPEC_X86_64) cam4_s_base	不通过	生成的ll过大, 存在bug	5
(SPEC_X86_64) cam4_s_base	不通过	生成的ll过大, 存在bug	5
(SPEC_X86_64) cam4_validate_527_base	不通过	生成的ll过大, 存在bug	5
(SPEC_X86_64) cam4_validate_627_base	不通过	生成的ll过大, 存在bug	5
(SPEC_X86_64) cam4_validate_627_base	通过		5
(SPEC_X86_64) cam4_validate_627_base	通过		5
(SPEC_X86_64) cpugcc_r_base	通过		5
(SPEC_X86_64) cpuxalan_r_base	通过		5
(SPEC_X86_64) deepsjeng_r_base	通过		5
(SPEC_X86_64) deepsjeng_s_base	通过		5
(SPEC_X86_64) diffwrf_521_base	通过		5
(SPEC_X86_64) diffwrf_621_base	通过		5
(SPEC_X86_64) exchange2_r_base	通过		5
(SPEC_X86_64) exchange2_s_base	通过		5
(SPEC_X86_64) fotonik3d_r_base	不通过	没有反汇编生成有效的C源文件, 也没有生成记录生成信息的json文件	5
(SPEC_X86_64) fotonik3d_s_base	通过		5
(SPEC_X86_64) imagevalidate_511_base	通过		5
(SPEC_X86_64) imagevalidate_525_base	通过		5
(SPEC_X86_64) imagevalidate_526_base	通过		5
(SPEC_X86_64) imagevalidate_538_base	通过		5
(SPEC_X86_64) imagevalidate_625_base	通过		5
(SPEC_X86_64) imagevalidate_638_base	通过		5
(SPEC_X86_64) imagick_r_base	通过		5
(SPEC_X86_64) imagick_s_base	通过		5
(SPEC_X86_64) lbm_r_base	通过		5

(SPEC_X86_64) lbm_s_base	通过		5
(SPEC_X86_64) ldecod_r_base	通过		5
(SPEC_X86_64) ldecod_s_base	通过		5
(SPEC_X86_64) leela_r_base	通过		5
(SPEC_X86_64) leela_s_base	通过		5
(SPEC_X86_64) mcf_r_base	通过		5
(SPEC_X86_64) mcf_s_base	通过		5
(SPEC_X86_64) nab_r_base	通过		5
(SPEC_X86_64) nab_s_base	通过		5
(SPEC_X86_64) namd_r_base	通过		5
(SPEC_X86_64) omnetpp_r_base	通过		5
(SPEC_X86_64) omnetpp_s_base	通过		5
(SPEC_X86_64) parest_r_base	通过		5
(SPEC_X86_64) perlbench_s_base	通过		5
(SPEC_X86_64) povray_r_base	通过		5
(SPEC_X86_64) roms_r_base	通过		5
(SPEC_X86_64) sgcc_base	通过		5
(SPEC_X86_64) specrand_fr_base	通过		5
(SPEC_X86_64) specrand_fs_base	通过		5
(SPEC_X86_64) specrand_ir_base	通过		5
(SPEC_X86_64) specrand_is_base	通过		5
(SPEC_X86_64) speed_bwaves_base	通过		5
(SPEC_X86_64) speed_pop2_base	不通过	没有反汇编生成有效的C源文件，也没有生成记录生成信息的json文件	5
(SPEC_X86_64) sroms_base	通过		5
(SPEC_X86_64) wrf_r_base	不通过	没有反汇编生成有效的C源文件，也没有生成记录生成信息的json文件	5
(SPEC_X86_64) wrf_s_base	不通过	没有反汇编生成有效的C源文件，也没有生成记录生成信息的json文件	5
(SPEC_X86_64) x264_r_base	通过		5
(SPEC_X86_64) x264_s_base	通过		5
(SPEC_X86_64) xz_r_base	通过		5
(SPEC_X86_64) xz_s_base	通过		5
(SPEC_AAArch64) bwaves_r_base	通过		5
(SPEC_AAArch64) cactusBSSN_r_base	通过		5

(SPEC_AAArch64) cpugcc_r_base	通过		5
(SPEC_AAArch64) exchange2_r_base	通过		5
(SPEC_AAArch64) exchange2_s_base	通过		5
(SPEC_AAArch64) fotonik3d_r_base	通过		5
(SPEC_AAArch64) fotonik3d_s_base	通过		5
(SPEC_AAArch64) lbm_r_base	通过		5
(SPEC_AAArch64) lbm_s_base	通过		5
(SPEC_AAArch64) mcf_r_base	通过		5
(SPEC_AAArch64) mcf_s_base	通过		5
(SPEC_AAArch64) nab_r_base	通过		5
(SPEC_AAArch64) nab_s_base	通过		5
(SPEC_AAArch64) roms_r_base	通过		5
(SPEC_AAArch64) sgcc_base	通过		5
(SPEC_AAArch64) specrand_fr_base	通过		5
(SPEC_AAArch64) specrand_fs_base	通过		5
(SPEC_AAArch64) specrand_ir_base	通过		5
(SPEC_AAArch64) specrand_is_base	通过		5
(SPEC_AAArch64) speed_bwaves_base	通过		5
(SPEC_AAArch64) speed_pop2_base	通过		5
(SPEC_AAArch64) sroms_base	通过		5
(SPEC_AAArch64) xz_r_base	通过		5
(SPEC_AAArch64) xz_s_base	通过		5
(SPEC_AAArch64) cactuBSSN_s_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AAArch64) cpuxalan_r_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AAArch64) deepsjeng_r_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AAArch64) deepsjeng_s_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AAArch64) leela_r_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AAArch64) leela_s_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5

(SPEC_AArch64) namd_r_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AArch64) omnetpp_r_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AArch64) omnetpp_s_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AArch64) parest_r_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AArch64) perlbench_r_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AArch64) perlbench_s_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5
(SPEC_AArch64) xalancbmk_s_base	不通过	[capstone2llvmir]: Arm64: extractVectorValue(): Unknown VESS type	5