

基于mugen的oERV 24.03 LTS自动化测试准备，实例学习mugen测试结果分析

目录

- 自动化测试准备的意义
- 当前oERV独立发版镜像测试结果
- 针对测试结果的分析
- 总结

自动化测试准备的意义

在oERV 2309测试中，对于不断迭代的测试镜像的测试中，由于准备不充分导致测试结果中经常出现误报漏报的情况，因此对于即将到来的oERV 24.03LTS版本，有必要进行一些自动化测试准备以应对较为快速的镜像迭代周期。

而在oERV 2309测试中，测试重点主要是BaseOS范围内的软件包，因此对应的测试用例也主要集中在这些软件包包含的范围内，而随着之后的oERV独立镜像发版工作的开展，需要对所有mugen内包含的测试用例进行测试，因此需要对接下来的测试工作进行一些准备，以应对较为紧张的测试周期。

在之前的测试中，主要的测试流程如下

- 在oERV上进行mugen测试
- 收集失败测试用例
- 在oE x86上进行验证mugen测试
- 将结果进行比对
- 分析测试用例结果

在当前测试流程中主要问题在于验证测试时需要我们自己在oE x86上进行测试来获取log来进行对比测试，对于时间较为紧张的测试周期来说很难在规定时间内完成。因此在后续测试前，有必要对2309 RV及x86进行全量测试，并根据测试结果提前做好准备，后续测试流程便可参照2309的测试结果进行对比，缩短分析时间

在oERV独立发版测试中，需要对qemu镜像进行mugen测试，因此在此测试中对测试结果进行详细的分析将有利于对后续的测试作一个充分的参考。

当前oERV独立发版镜像的测试结果

在当前的oERV独立发版镜像中，在qemu镜像上测试了mugen上游仓库中包含的5280个测试用例，同时在oE 2309 x86 qemu镜像上进行了相同数量测试用例的对比测试，当前结果如下

在BaseOS范围内

- 47个测试用例在RISCV上失败，在x86上成功
- 311个测试用例在RISCV及x86上均失败
- 1725个测试用例在RISCV及x86上均成功

在BaseOS范围外

- 136个测试用例在RISCV上失败，在x86上成功
- 704个测试用例在RISCV及x86上均失败
- 1742个测试用例在RISCV及x86上均成功

测试结果的分析

mugen测试失败原因很多，除了真正能发现问题的失败测试用例外，许多测试用例都是因为非软件包问题导致的失败且数量占绝大多数，因此对于测试结果的分析就尤为重要，对于那些非软件包导致的问题来说，主要分为以下三类：

- 测试用例不兼容RISCV架构
- 测试用例本身存在问题
- qemu镜像本身存在效率问题导致测试失败
- 测试环境问题导致测试失败

测试用例不兼容RISCV架构

不少测试用例在编写时仅考虑了单一架构的情况，因此在编写时预期结果仅考虑了x86架构的结果，例如测试用例oe_test_MEMinfo_001：

```
+ lshw -c memory  
+ grep bank -A 5  
+ grep description:  
+ CHECK_RESULT 1
```

lshw -c memory命令在x86上输出如下

```
*-memory
  description: System Memory
  physical id: 1000
  size: 4GiB
  capabilities: ecc
  configuration: errordetection=multi-bit-ecc
*-bank
  description: DIMM RAM
  vendor: QEMU
  physical id: 0
  slot: DIMM 0
  size: 4GiB
```

而在RISCV上输出结果如下

```
*-memory
  description: System memory
  physical id: 9
  size: 7930MiB
```

因此RV并不适用于该测试用例，在测试结果中归为非问题

而部分测试用例虽然在RISCV和x86上均失败，但是失败结果并不相同，例测试用例
oe_test_yasm_01

x86与RISCV报错均相同

```
+ ld hello.o -o hello  
oe_test_yasm_01.sh: line 55: ld: command not found
```

该测试用例主要使用yasm编译一个x86的asm文件，并执行已检验yasm的正确性，但在riscv架构中依然编译x86的asm文件，因此ld阶段就会报错，并且之后的运行也是不合理的，故此测试用例也不兼容RISCV架构

对于不兼容RISCV架构的测试用例后续处理

对于不兼容的测试用例，一方面可以通过提交issue来使得测试用例兼容RISCV，例如上文提到的oe_test_yasm_01便可通过提交issue新增支持riscv架构的yasm汇编文件的方式来兼容RISCV架构，另一方面，对于一些完全不适合RISCV的测试用例，在后续的测试中可以自动将这些测试用例的失败结果归为非问题，以减少重复工作量

测试用例本身存在问题

由于部分测试用例未进行更新或测试软件源本身对功能做出了修改，因此这些测试用例在任何架构上测试均会失败，存在的相同错误情况可能有如下情况：

测试用例在编写时没有考虑测试时的环境，因此缺少如所需软件包的安装命令等操作导致测试用例失败

例测试用例oe_test_bison，由于在环境阶段未安装gcc导致测试用例在RV及x86上均失败

```
+ gcc -std=c99 -o set_calc set_calc.tab.c lex.yy.c
oe_test_bison.sh: line 49: gcc: command not found
+ test -f set_calc
```

测试用例进行测试的命令在新版本软件包上已经废弃或更改导致测试无法继续

例测试用例oe_test_openssl_generate_key_pair,此版本生成的密钥已不为PKCS#1 格式

```
LOG_INFO "Start to run test."  
openssl genrsa -out server.pri.1024  
CHECK_RESULT $?  
grep 'BEGIN RSA PRIVATE KEY' server.pri.1024
```

测试用例本身编写逻辑存在问题导致测试环境稍微变更后便失败

例测试用例oe_test_amanda_aespipe，从测试用例网站上下载的为aespipe-v2.4g 而测试用例编写的为aespipe-v2.4f，导致测试用例失败

```
+ tar -xvf aespipe-latest.tar.bz2
aespipe-v2.4g/
....
aespipe-v2.4g/sha512.h
+ cd aespipe-v2.4f
oe_test_amanda_aespipe.sh: line 27: cd: aespipe-v2.4f: No such file or directory
```

处理办法

对于这些测试用例，在明确为RV和x86问题相同后可归为非问题并暂时忽略，同时部分测试用例或软件包的错误可以通过向对应的仓库提交issue来进行反馈。

qemu镜像本身存在效率问题导致测试失败

此现象在2309独立发版镜像上大量出现，原因在于该镜像版本运行速度过慢，导致不少在测试用例在执行完SLEEP_WAIT命令后部分命令仍未执行完毕导致测试出错，例测试用例oe_test_lxc_unshare_update 由于qemu运行速度过慢导致测试用例编写的sleep wait 2之后上一个命令仍未完成导致测试失败，修改等待时间为20秒后测试用例即可通过

另一个例子为vdo测试套，该测试套主要测试vdo功能，因此会进行大量数据的测试（10G左右），对于常规的qemu镜像来说有很大概率在单个测试用例执行命令超过mugen设置的时间（即30分钟）后仍未结束导致测试失败，因此测试套在进行批量测试时非常容易超时导致失败。

解决办法

等待后续更高效率的qemu镜像更新后对这些样例进行重测，对于vdo等测试套，单独启用镜像进行复测即可。

测试环境问题导致测试失败

由于测试采用测试小队编写的多线程测试脚本进行测试，而测试脚本以测试套为单位创建qemu虚拟机进行测试，因此在测试某些测试用例数量过大的测试套中，会出现部分测试用例环境无法完全复原导致测试环境损坏的情况，使得测试套中后续的测试用例无法继续测试从而导致失败，而部分测试用例也会影响多线程测试脚本创建的环境从而导致测试失败。

解决办法

当前针对环境问题导致失败的测试用例目前的处理办法是单独打开一个qemu虚拟机进行单独测试，后续测试小队考虑修改多线程测试脚本使得脚本更加稳定，同时也在考虑将测试用例数量过大的测试套拆分为多个小的测试用例集合进行测试，最大程度避免测试环境的影响

总结

测试结果的对比分析非常重要，也需要大量的时间进行投入，因此提前做好当前版本的测试结果分析对于后续开展对比分析而言非常重要，能为后续工作省下大量时间及精力，保证测试结果的准确性的同时加快镜像的迭代周期。

谢谢大家