实验报告

课程名称：软件测试

实验名称：KLEE 实验课

专业班级：软件工程 16 级四班

学　　号：1611736

姓　　名：钟腾

2018 年　12 月　11 日

实验一

实验名称	KLEE 实验课		
实验地点	泰达 5 区 105	实验时间	2018/12/11

实验目的和要求

1）熟悉 Linux 系统使用方式;

2）了解 LLVM 的架构和使用方式;

3）了解 Docker 使用

4）了解 Github 使用方式

5）了解并掌握基于约束的自动化测试工具的基本原理和使用。

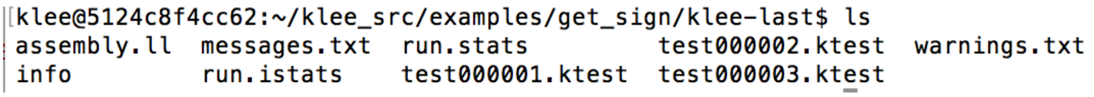
实验环境

Docker 2.0.0.0
Mac os 10.13.5

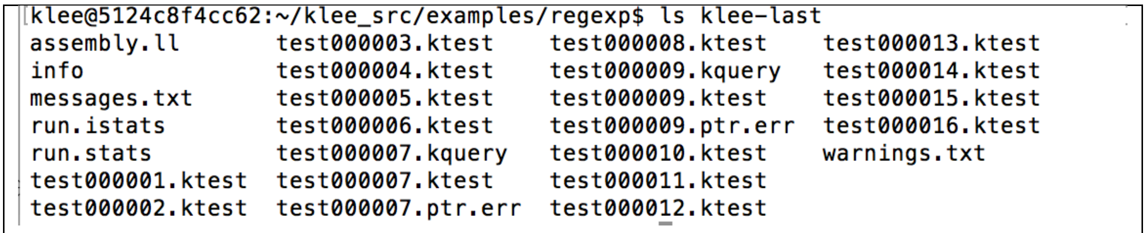
实验过程

1、一开始在 mac os 中使用 sudo apt-get 安装 KLEE 失败，后来发现 mac os 没有 apt-get 功能，于是采用 docker 安装 KLEE 镜像。

2、完成 tutorial 1 和 tutorial 2 ：

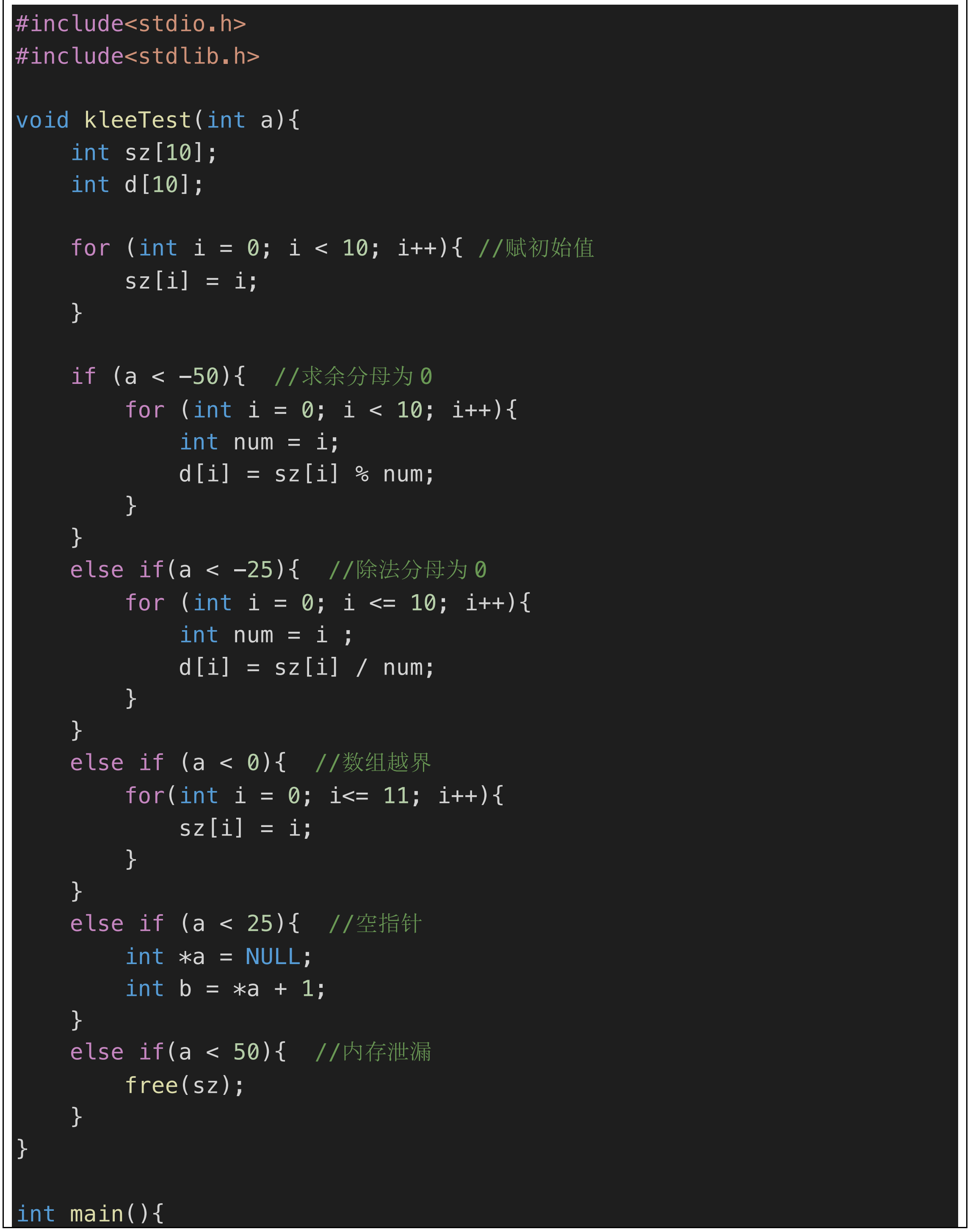


```
klee@5124c8f4cc62:~/klee_src/examples/get_sign/klee-last$ ls
assembly.ll  messages.txt  run.stats         test000002.ktest  warnings.txt
info         run.istats    test000001.ktest  test000003.ktest
```

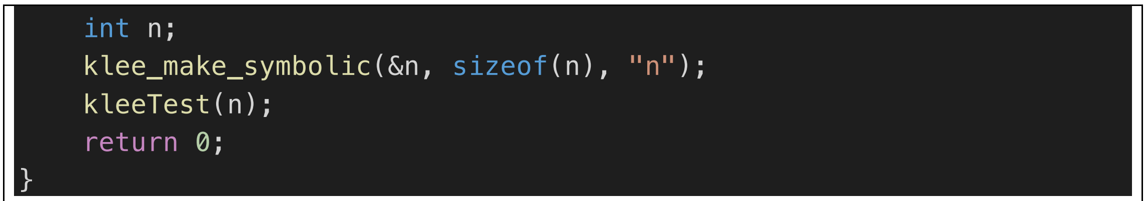


```
klee@5124c8f4cc62:~/klee_src/examples/regexp$ ls klee-last
assembly.ll       test000003.ktest   test000008.ktest   test000013.ktest
info              test000004.ktest   test000009.kquery  test000014.ktest
messages.txt      test000005.ktest   test000009.ktest   test000015.ktest
run.istats        test000006.ktest   test000009.ptr.err test000016.ktest
run.stats         test000007.kquery  test000010.ktest   warnings.txt
test000001.ktest  test000007.ktest   test000011.ktest
test000002.ktest  test000007.ptr.err test000012.ktest
```

3、用 c 语言编写缺陷代码如下：

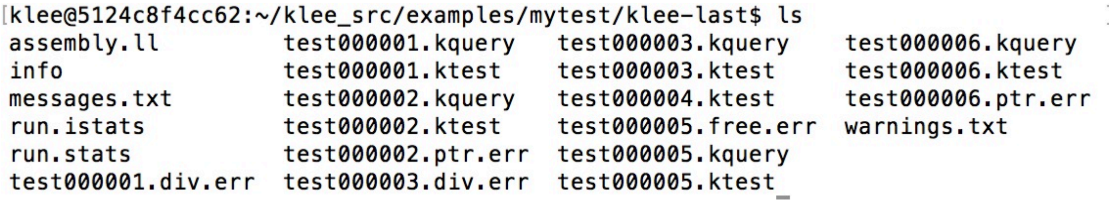


```c
#include<stdio.h>
#include<stdlib.h>

void kleeTest(int a){
    int sz[10];
    int d[10];

    for (int i = 0; i < 10; i++){ //赋初始值
        sz[i] = i;
    }

    if (a < -50){  //求余分母为 0
        for (int i = 0; i < 10; i++){
            int num = i;
            d[i] = sz[i] % num;
        }
    }
    else if(a < -25){  //除法分母为 0
        for (int i = 0; i <= 10; i++){
            int num = i ;
            d[i] = sz[i] / num;
        }
    }
    else if (a < 0){  //数组越界
        for(int i = 0; i<= 11; i++){
            sz[i] = i;
        }
    }
    else if (a < 25){  //空指针
        int *a = NULL;
        int b = *a + 1;
    }
    else if(a < 50){  //内存泄漏
        free(sz);
    }
}

int main(){
```



```
    int n;
    klee_make_symbolic(&n, sizeof(n), "n");
    kleeTest(n);
    return 0;
}
```

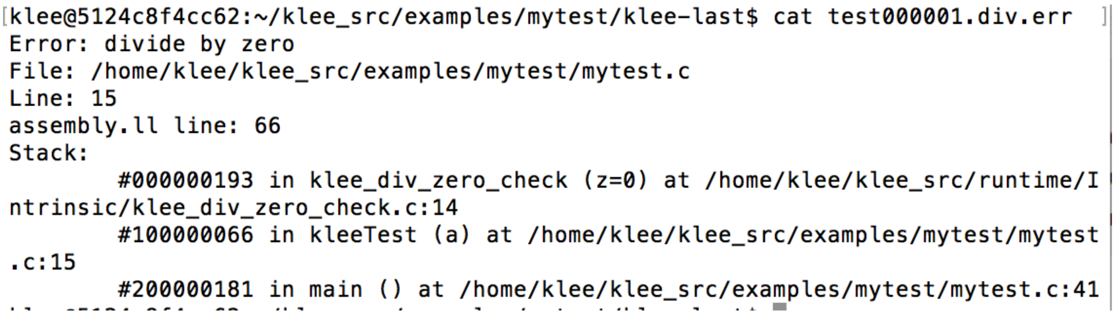
生成文件截图：



```
klee@5124c8f4cc62:~/klee_src/examples/mytest/klee-last$ ls
assembly.ll          test000001.kquery   test000003.kquery    test000006.kquery
info                 test000001.ktest    test000003.ktest     test000006.ktest
messages.txt         test000002.kquery   test000004.ktest     test000006.ptr.err
run.istats           test000002.ktest    test000005.free.err  warnings.txt
run.stats            test000002.ptr.err  test000005.kquery
test000001.div.err   test000003.div.err  test000005.ktest
```

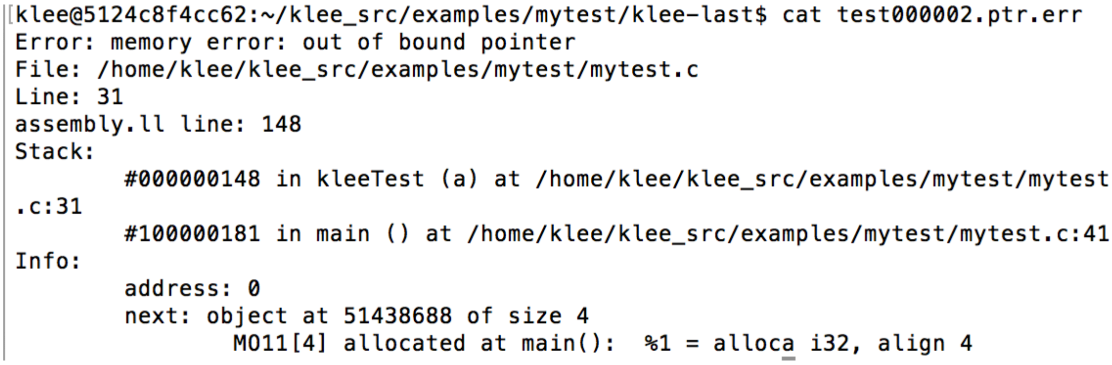
错误文件内容：

1、分母为 0



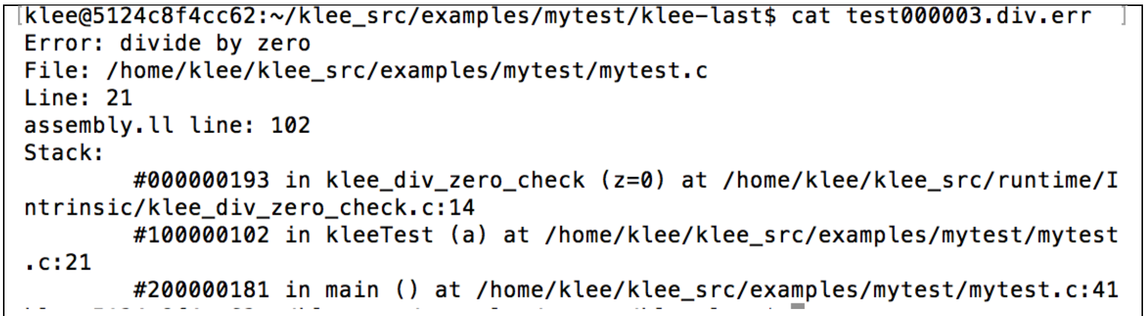
```
klee@5124c8f4cc62:~/klee_src/examples/mytest/klee-last$ cat test000001.div.err
Error: divide by zero
File: /home/klee/klee_src/examples/mytest/mytest.c
Line: 15
assembly.ll line: 66
Stack:
        #000000193 in klee_div_zero_check (z=0) at /home/klee/klee_src/runtime/I
ntrinsic/klee_div_zero_check.c:14
        #100000066 in kleeTest (a) at /home/klee/klee_src/examples/mytest/mytest
.c:15
        #200000181 in main () at /home/klee/klee_src/examples/mytest/mytest.c:41
```

2、空指针



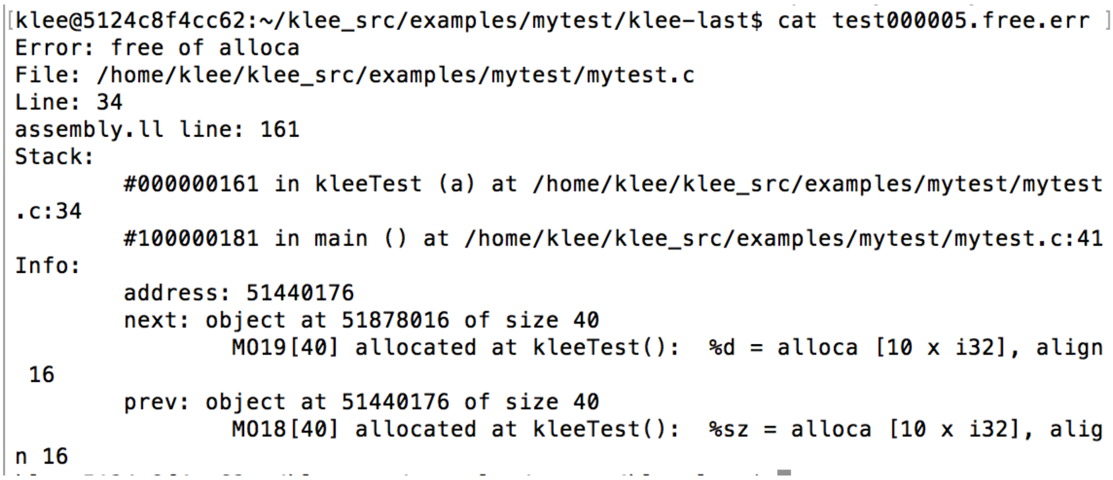
```
klee@5124c8f4cc62:~/klee_src/examples/mytest/klee-last$ cat test000002.ptr.err
Error: memory error: out of bound pointer
File: /home/klee/klee_src/examples/mytest/mytest.c
Line: 31
assembly.ll line: 148
Stack:
        #000000148 in kleeTest (a) at /home/klee/klee_src/examples/mytest/mytest
.c:31
        #100000181 in main () at /home/klee/klee_src/examples/mytest/mytest.c:41
Info:
        address: 0
        next: object at 51438688 of size 4
                MO11[4] allocated at main():  %1 = alloca i32, align 4
```

3、分母为 0



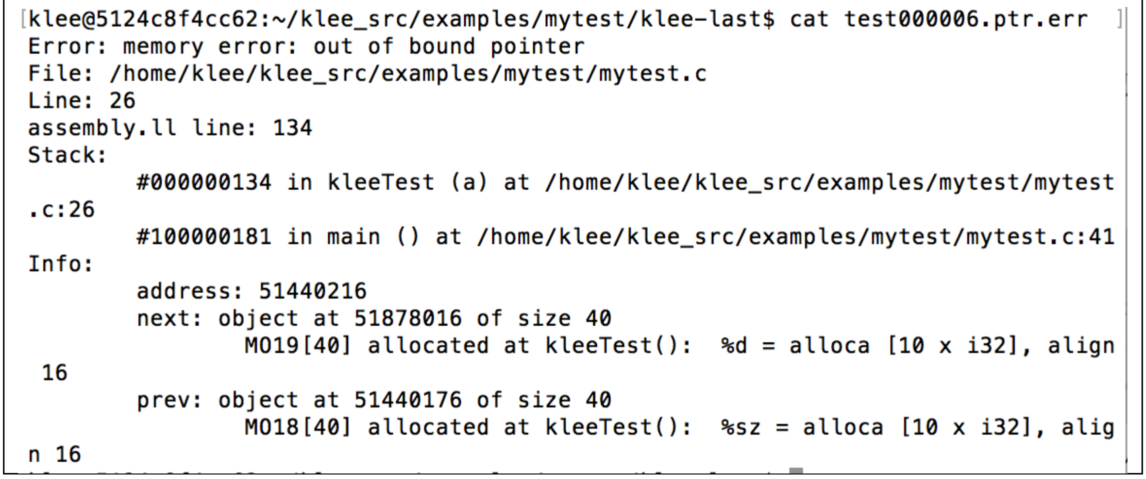
```
klee@5124c8f4cc62:~/klee_src/examples/mytest/klee-last$ cat test000003.div.err
Error: divide by zero
File: /home/klee/klee_src/examples/mytest/mytest.c
Line: 21
assembly.ll line: 102
Stack:
        #000000193 in klee_div_zero_check (z=0) at /home/klee/klee_src/runtime/I
ntrinsic/klee_div_zero_check.c:14
        #100000102 in kleeTest (a) at /home/klee/klee_src/examples/mytest/mytest
.c:21
        #200000181 in main () at /home/klee/klee_src/examples/mytest/mytest.c:41
```

4、内存泄漏



```
klee@5124c8f4cc62:~/klee_src/examples/mytest/klee-last$ cat test000005.free.err
Error: free of alloca
File: /home/klee/klee_src/examples/mytest/mytest.c
Line: 34
assembly.ll line: 161
Stack:
        #000000161 in kleeTest (a) at /home/klee/klee_src/examples/mytest/mytest
.c:34
        #100000181 in main () at /home/klee/klee_src/examples/mytest/mytest.c:41
Info:
        address: 51440176
        next: object at 51878016 of size 40
                MO19[40] allocated at kleeTest():  %d = alloca [10 x i32], align
 16
        prev: object at 51440176 of size 40
                MO18[40] allocated at kleeTest():  %sz = alloca [10 x i32], alig
n 16
```

5、数组越界



```
klee@5124c8f4cc62:~/klee_src/examples/mytest/klee-last$ cat test000006.ptr.err
Error: memory error: out of bound pointer
File: /home/klee/klee_src/examples/mytest/mytest.c
Line: 26
assembly.ll line: 134
Stack:
        #000000134 in kleeTest (a) at /home/klee/klee_src/examples/mytest/mytest
.c:26
        #100000181 in main () at /home/klee/klee_src/examples/mytest/mytest.c:41
Info:
        address: 51440216
        next: object at 51878016 of size 40
                MO19[40] allocated at kleeTest():  %d = alloca [10 x i32], align
 16
        prev: object at 51440176 of size 40
                MO18[40] allocated at kleeTest():  %sz = alloca [10 x i32], alig
n 16
```

心得体会
本次实验学会了使用docker和klee工具，并学会了如何编写带有软件缺陷的程序，对于掌握软件测试的各种方法具有很大帮助。鉴于国内外对于klee工具使用的说明介绍较少，因为将本次作业作为了技术帖发布在了csdn等网站。对于掌握新知识，我感到很兴奋。