# 差分测试实验说明

November 29, 2022

## 1 实验背景

差分测试是软件测试中的常用技术。差分测试的目标是：对于两个实现相同功能的程序，构造特殊的输入，使得两个程序的行为不同。

形式化地，对于两个程序 program1 和 program2，给定同一符合输入数据格式约定的输入文件 input，我们记 output1 是 program1 的输出，output2 是 program2 的输入，当 output1 与 output2 不同时，即认为测试成功。

## 2 实验任务

本次实验由两个子任务 $Mission_A$,$Mission_B$ 构成，在每个子任务中，都包含两个程序 program1,program2，这两个程序解决的是同一个问题，但因为种种原因，它们接受某些输入后有着不同的行为。

你的任务是构造一组输入 input，满足特定的要求 (在每个 $Mission$ 中的 specification.txt 中给出)，使得 program1,program2 在输入 input 后在规定时间限制出现了不同的行为。这里不同的行为可以是以下任意一种:

- ◇ 两个程序都正常结束，但是输出不同
- ◇ 一个程序发生崩溃，而另外一个程序没有
- ◇ 一个程序没有在规定时间限制中结束，而另外一个程序正常结束

此外，program1 与 program2 在读入时都通过调用 libobfuscate.so 动态库中的混淆函数进行了特别处理，我们保证你提供的**合法**输入在经过该混淆函数处理后，仍然符合 specification.txt 中的数据要求。

## 3 下发文件说明

每一个 Mission 都包含以下文件

### program1.c,program2.c

解决同一问题的两个程序，保证这两个程序没有语法问题。

### specification.txt

- ◇ 规定了程序接受的输入格式
- ◇ 规定了程序的运行时间限制

### userdef.h

包含混淆函数声明的头文件

### libobfuscate.so

一个动态库文件，其中实现了混淆函数 $F$。形式化地来说，我们设程序接受的输入集合为 $S$, 那么 $F$ 实际上是一个映射函数且满足 $F : S \to S_0$, $s.t.\ S_0 \subseteq S$

**Makefile**

用于编译 program1.c,program2.c 生成可执行文件

# 4 编译、运行及测试

下面的操作都基于上述下发文件在同一目录下，且终端进入了该目录下

- ◇ 指令 make 进行编译生成可执行文件 program1,program2
- ◇ 指令 make clean 用于清除生成的可执行文件以及中间代码
- ◇ 将你构造好的输入写入当前目录下的 input 文件，运行 make check 即可判断差分测试是否成功

# 5 提交说明

## 5.1 提交内容

应当包含**实验报告**，每个 Mission 的 **input 文件**以及**相关的源程序**，实验报告中应当包含:

- ◇ 小组的分工情况
- ◇ 生成 input 文件的策略；如果编写了相关代码文件，可以一并提交，但请说明程序的具体功能
- ◇ 实验报告不鼓励长篇大论，说清楚即可

## 5.2 评分标准

- ◇ $Mission_A$，$Mission_B$ 各占 7.5'
- ◇ 对于每个 $Mission$: 找到符合要求的 input 文件占 60%，实验报告占 40%
- ◇ 按时提交
- ◇ 如果无法找到满足要求的输入文件，亦可在实验报告中说明你们所遇到的困难及所做的尝试，视具体情况酌情给分