链滴

程序切片技术

作者：Hanseltu

原文链接：https://ld246.com/article/1555083057303

来源网站：链滴

原文链接 程序切片技术

程序依赖的痛处

我们知道，程序之间的依赖关系相当复杂，分析起来十分麻烦，效率也不高，但是如果我们只分析我感兴趣的部分会怎么样呢？程序切片技术就是一种让我们只分析感兴趣部分的技术。

程序切片的定义

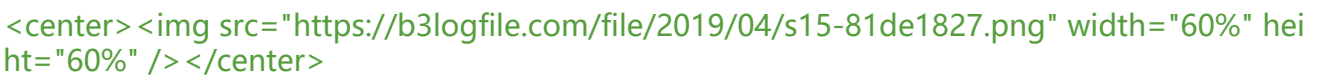
Mark Weiser 在 1979 年提出切片技术，应用于程序调试工作。在程序调试时， 程序员经常面临这的问题，他们发现程序中某个位置的某个变量的值出错了， 此时需要追踪哪些代码能够影响这个值的算。Mark Weiser 将这样的程序位置 n 和某个变量 v 的二元组<n,v>定义为切片标准，程序中影响片标准的所有部分被称为程序切片。程序位置 n 一般用代码行号指定，而 v 可以是单一变量也可 以变量集合。

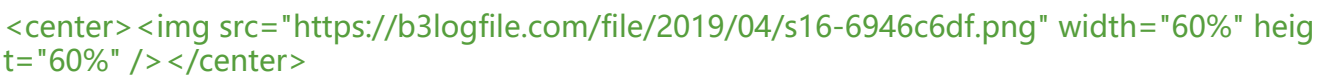
如上图，与 print(i)有关的代码被标记出来了。

程序切片的应用场景

- 调试
- 测试
- 逆向工程
- 优化
- 程序剖析
- 恶意代码分析

程序切片方法





程序切片技术分类

- 静态切片 vs 动态切片
- 前向切片 vs 后向切片
- 可执行切片 vs 不可执行切片
- 基于边的切片 vs 基于点的切片

静态切片技术

- 优势
  - 考虑了所有的执行路径
  - 快速计算
  - 节省空间
- 缺点
  - 可能会有路径重复
  - 指针指向不准确

动态切片技术

- 优势
  - 精确地考虑单次执行
  - 没有别名或多路径的不精确
  - 包含更少的程序语句
- 缺点
  - 捕获跟踪并计算DDG代价较高（跟踪文件通常达到GB量级）
  - 计算缓慢（浪费了大量内存）
  - 很多语句实例和动态依赖性要检查

其他切片方法

>> 回到课程主目录