BugVisualizer使用说明

本功能是对目标源代码文件（或者是目标项目）完成检测之后，将测试报告以网页的方法清晰的展示，以便于用户理解。其操作方法如下：

1）进入BugVisualizer的所在目录下，打开终端；

2）在终端下执行命令：

java –jar BugVisualizer.jar –report=[REPORT\_PATH] –project=[PROJECT\_PATH] –web=[WEB\_PATH]

其中：

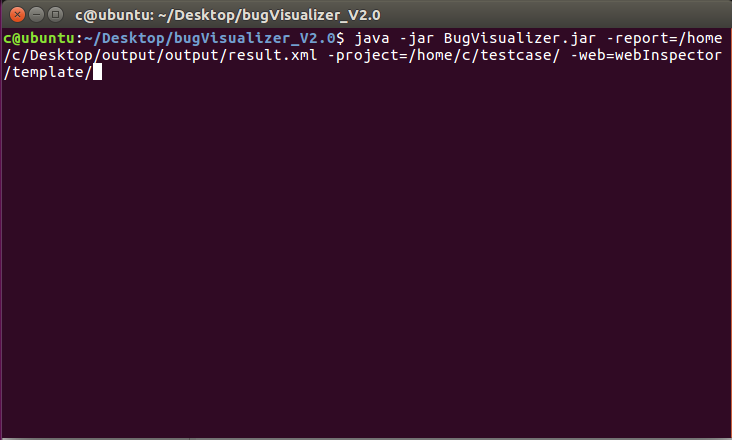
第一项–report=[REPORT\_PATH]是必需参数，REPORT\_PATH指定了检测结果报告的存放目录，一般为输出目录output下的result.xml文件；

第二项-project=[PROJECT\_PATH]是必需参数，PROJECT\_PATH指定了被检测文件(或者被检测项目)所存放的文件夹的路径；此路径是源文件的base路径，源文件的绝对路径减去base路径，就是生成的源文件相对路径，以后将源文件拷贝到其他地方时，一定要拷贝与相对路径一致的目录结构，比如相对路径是a/b/test.c,那么一定要拷贝到目录a;

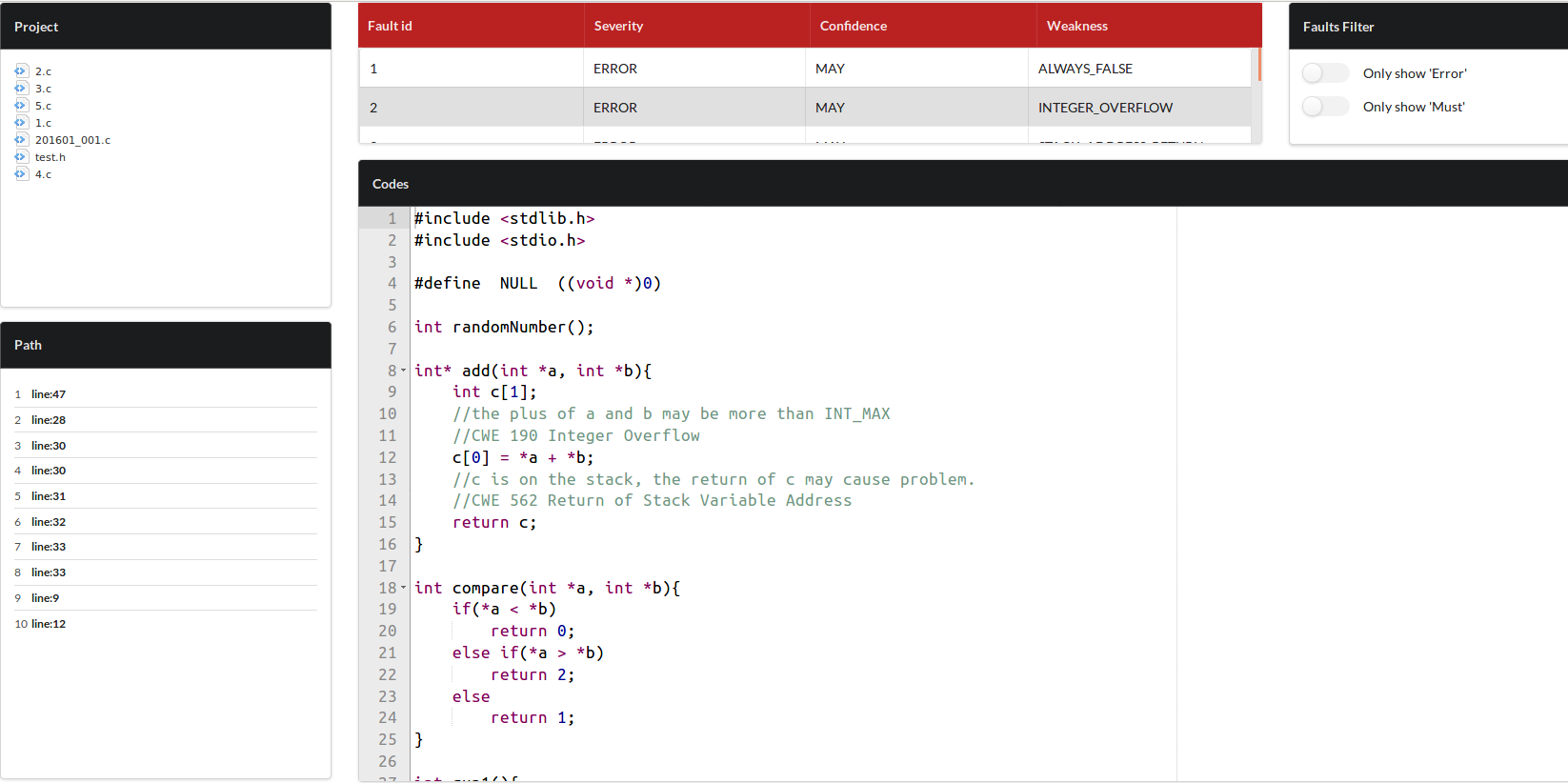
第三项-web=[WEB\_PATH]是必需参数，WEB\_PATH指定了用于展示检测结果的网页的模板的路径。BugVisualizer提供了一个自带的模板，其存放于BugVisualizer目录下的webInspector/template文件夹。

**使用示例：**

打开BugVisualizer文件夹，并在这个目录下打开终端执行，下列命令：



其中-report后紧接的是测试结果的汇总报告的存放目录。-project后的参数是被检测文件所存放的文件夹。-web则是BugVisualizer使用的默认模板。执行该命令之后，会自动打开浏览器，以网页的方式展示出测试结果，如下图所示：



左侧一列中，第一栏Project展示了该目录下的所有C语言文件。第二栏Path展示了触发所选择的缺陷的执行流程，并且单击其中的任意一行，右侧的代码框将自动跳转到对应位置。

中央上方一栏展示了检测的所有的缺陷，给出了缺陷的编号，严重度，可能性，缺陷类别。选择其中任意一个缺陷，Path栏和Code栏会进行对应的跳转。

下方一栏展示了被测试文件的代码。

右上方一栏提供了过滤器，允许筛选缺陷的类型。

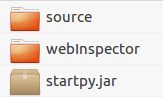
例如我们现在想关注函数返回栈上的临时变量的问题，可以选择第3个缺陷id，此时左下角的path路径会更新。新的path路径说明了触发这个问题时，整个程序的执行流程。我们再选中最后一行，这样可以得到直接触发问题的语句所在地点。

(2)BugVisualizer迁移使用

小节（1）展示了在开发人员（自己）的电脑上运行BugVisualizer查看检查结果的方法，当要在其他电脑上查看结果时需要完成以下步骤：

(1)需要拷贝到其他机器上的文件或目录包括：BugVisualizer目录下的{startPy.jar文件 和 webInspector目录，以及源程序目录}。

*注：*对于源程序目录结构的说明：假设在小节（1）中PROJECT\_PATH = /1/2/3/，源文件的绝对路径/1/2/3/a/b/test.c。那么生成的相对路径为a/b/test.c。所以拷贝源程序到其他电脑上是应该拷贝a/b/test.c这样的目录结构。可以将a/b/test.c这样的结构放到source目录里.最后拷贝到其他电脑上的目录结构如下图：



(2)进入到上图显示的同级目录中：

运行：java -jar startpy.jar source即可。当然source的位置可以改，但是在命令行中要指定到source。

或者进入到webInspector/template目录中运行：python startInspect.py xxx/source(一定要指定source的目录）。

1.目录结构

--bin

java可执行文件

--webInspector

--template

--Ace -- web code editor

--semantic -- semantic UI 前端布局

--jqueryFileTree -- 文件树的js和样式表

--js -- 全局的控制文件main.js

--css -- 层叠样式表文件, main.css中设置了Marker和active path的样式

--data -- data.js, 包含了项目文件的路径和缺陷路径报告的内容。

--index.html -- 索引html

--startInspect.py -- 一个python2脚本，会创建一个本地的httpserver，以处理带参数的get请求（dir、file）。执行后自动打开浏览器

--test

用于放置示例的report.xml的文件夹，可以是其他位置

--lib

一些jar包

--src

java源码

--README.txt

本文件

2.main.js代码的说明

initControl() --入口函数

initEditor() --初始化editor，设置一些属性

initFileTree() --初始化文件树，绑定文件路径和点击事件

initFaultsSet() --读入缺陷路径，并根据筛选要求进行筛选，添加点击事件

initFaultsFilter() --定义了checkbox的onchange事件

initPath() --将path的内容清空

loadFile() --根据路径得到代码，设置代码，去除以前的所有markers和annotations

loadFaultPath() --fault表格里元素的点击事件。加载一条缺陷路径，并默认打开第一个节点的文件

removeOldMarkers() --去除所有之前的markers

addMarker() --去掉之前active元素的样式，触发点击事件的元素加上active样式，加载文件，加上annotation和marker, scroll到marker的行

3.java部分

主要功能就是将得到工程文件的路径和report xml的信息，写到data.js中