

gcc 参数 -fprofile-arcs -ftest-coverage作用

原创 Walter_Jia 于 2013-03-27 16:29:34 发布 8581 收藏 9

版权

分类专栏: Linux Develop



Linux Develop 专栏收录该内容

2 订阅

135 篇文章

订阅专栏

gcov 是一个可用于C/C++的代码覆盖工具，是gcc的内建工具。下面介绍一下如何利用gcov来收集代码覆盖信息。想要用gcov收集代码覆盖信息，需要在gcc编译代码的时候加上这2个选项“-fprofile-arcs -ftest-coverage”，把这个简单的程序编译一下

```
gcc -fprofile-arcs -ftest-coverage hello.c -o hello
```

编译后会得到一个可执行文件hello和hello.gcno文件，当用gcc编译文件的时候，如果带有“-ftest-coverage”参数，就会生成这个.gcno文件，它包含了程序块和行号等信息
接下来可以运行这个hello的程序

```
./hello 5  
./hello 12
```

运行结束以后会生成一个hello.gcda文件，如果一个可执行文件带有“-fprofile-arcs”参数编译出来，并且运行过至少一次，就会生成。这个文件包含了程序基本块跳转的信息。接下来可以用gcov生成代码覆盖信息：

```
gcov hello.c
```

运行结束以后会生成2个文件hello.c.gcov和myfunc.c.gcov。打开看里面的信息：

```
-: 0:Source:myfunc.c  
-: 0:Graph:hello.gcno  
-: 0:Data:hello.gcda  
-: 0:Runs:1  
-: 0:Programs:1  
-: 1:#include  
-: 2:  
-: 3:void test(int count)  
1: 4:{  
-: 5: int i;  
10: 6: for (i = 1; i < count; i++)  
-: 7: {  
9: 8: if (i % 3 == 0)  
3: 9: printf ("%d is divisible by 3 /n", i);  
9: 10: if (i % 11 == 0)  
#####: 11: printf ("%d is divisible by 11 /n", i);  
9: 12: if (i % 13 == 0)  
#####: 13: printf ("%d is divisible by 13 /n", i);  
-: 14: }  
1: 15:}
```

被标记为#####的代码行就是没有被执行过的，代码覆盖的信息是正确的，但是让人去读这些文字，实在是一个杯具。不用担心，有另外一个工具叫lcov，可以用程序解析这些晦涩的字符，最终输出成html格式的报告，很好吧！

```
lcov -d . -t 'Hello test' -o 'hello_test.info' -b . -c
```

指定lcov在当前目录“.”去找代码覆盖的信息，输出为'hello_test.info'，这个hello_test.info是一个中间结果，需要把它用genhtml来处理一下，genhtml是lcov里面的一个工具。

```
genhtml -o result hello_test.info
```

指定输出目录是result。一个完整的html报告就生成了，做一个连接，把这个目录连到随便一个web server的目录下，就可以看报告了。

