

CS 350
Recitation Topic
Date: Sept. 6, 2005

Objective: use one basic testing support tool.

Topics:

1. Concept of test coverage
2. Documentation sources
3. Compiling and running
4. Assignment
5. Due dates

1. Concept: Common testing goals:

Statement coverage: run enough tests so that, at the minimum, every statement in the software is executed at least once. The clerical task of keeping track of this is labor-intensive and error-prone for big software systems.

Branch coverage: run enough tests so that all branches are taken at least once.

Hence tools like **gcov** (free version of tcov from the gnu project).

2. gcov documentation:

Three options:

1. `gcov -help`, short description of parameters
2. `info gcov`, multipage description; contains examples
3. `man gcov`, older version of info; less interactive

3. Compile and run instructions:

```
$ g++ -fprofile-arcs -ftest-coverage tmp.cpp
$ a.out
$ gcov tmp.cpp
90.00% of 10 source lines executed in file tmp.cpp
Creating tmp.cpp.gcov.
```

Some comments: the usual situation when testing is the program may need to be runs several times with different data or different conditions. Thus the gcov software is set up so that each run of the code updates coverage data so that the `*.gcov` file can be checked to see the cumulative effect of multiple test runs. For many testers, the contents of the `*.gcov` file is used to guide the creation of additional test files or additional test conditions.

Contents of output file tmp.c.gcov:

```
-: 0:Source:tmp.c
-: 0:Object:tmp.bb
-: 1:#include <stdio.h>
-: 2:
-: 3:int main (void)
1: 4:{
1: 5: int i, total;
-: 6:
1: 7: total = 0;
-: 8:
11: 9: for (i = 0; i < 10; i++)
10: 10:     total += i;
-: 11:
1: 12: if (total != 45)
#####: 13:     printf ("Failure\n");
-: 14: else
1: 15:     printf ("Success\n");
1: 16: return 0;
1: 17:}
```

If run with "gcov -b tmp.c" (to get branch coverage statistics):

```
$ gcov -b tmp.c
90.00% of 10 source lines executed in file tmp.c
80.00% of 5 branches executed in file tmp.c
80.00% of 5 branches taken at least once in file tmp.c
50.00% of 2 calls executed in file tmp.c
Creating tmp.c.gcov.
```

4. Assignment:

1. Use gcov to help test your first CS 350 program. You should test until you achieve 100% statement coverage and 100% branch coverage. This may require that you run the code multiple times. You can use the gcov report to guide you in selecting additional test cases.
2. Find the prices of two commercially available test coverage tools. Include source of your information (I search with google for "test coverage tool").

5. Due date:

With final submission of program 1.