Sun Studio 12: Fortran Programming Guide

Previous: 8.2 The time Command **Next**: Chapter 9 Performance and Optimization

8.3 The `tcov` Profiling Command

The `tcov` (1) command, when used with programs compiled with the `-xprofile=tcov` option, produces a statement-by-statement profile of the source code showing which statements executed and how often. It also gives a summary of information about the basic block structure of the program.

Enhanced statement level coverage is invoked by the `-xprofile=tcov` compiler option and the `tcov -x` option. The output is a copy of the source files annotated with statement execution counts in the margin.

Note –

The code coverage report produced by `tcov` will be unreliable if the compiler has inlined calls to routines. The compiler inlines calls whenever appropriate at optimization levels above `-O3`, and according to the `-inline` option. With inlining, the compiler replaces a call to a routine with the actual code for the called routine. And, since there is no call, references to those inlined routines will not be reported by `tcov`. Therefore, to get an accurate coverage report, do not enable compiler inlining.

8.3.1 Enhanced `tcov` Analysis

To use `tcov`, compile with `-xprofile=tcov`. When the program is run, coverage data is stored in *program* `.profile/tcovd`, where *program* is the name of the executable file. (If the executable were `a.out`, `a.out.profile/tcovd` would be created.)

Run `tcov -x` *dirname source_files* to create the coverage analysis merged with each source file. The report is written to *file* `.tcov` in the current directory.

Running a simple example:

```
demo%  f95 -o onetwo -xprofile=tcov one.f two.f
demo%  onetwo
       ... output from program
demo%  tcov -x onetwo.profile one.f two.f
demo%  cat one.f.tcov two.f.tcov
                      program one
      1 ->            do i=1,10
     10 ->                  call two(i)
                      end do
      1 ->            end
       .....etc
demo%
```

Environment variables `$SUN_PROFDATA` and `$SUN_PROFDATA_DIR` can be used to specify where the intermediary data collection files are kept. These are the `*.d` and `tcovd` files created by old and new style `tcov`, respectively.

These environment variables can be used to separate the collected data from different runs. With these variables set, the running program writes execution data to the files in `$SUN_PROFDATA_DIR/$SUN_PROFDATA/`.

Similarly, the directory that `tcov` reads is specified by `tcov -x $SUN_PROFDATA`. If `$SUN_PROFDATA_DIR` is set, `tcov` will prepend it, looking for files in `$SUN_PROFDATA_DIR/$SUN_PROFDATA/`, and not in the working directory.

Each subsequent run accumulates more coverage data into the `tcovd` file. Data for each object file is zeroed out the first time the program is executed after the corresponding source file has been recompiled. Data for the entire program is zeroed out by removing the `tcovd` file.

For the details, see the `tcov` (1) man page.

Previous: 8.2 The time Command

Next: Chapter 9 Performance and Optimization