Test coverage shown live on a Android mobile phone

Roland Bär

Verifysoft Technology GmbH, Offenburg, Germany

Droidcon, Berlin, March 2011

Outline

Why test coverage is required?

This is the only proof, that shows that you have tested anything. . .

Testwell CTC++

- A proprietary tool from Testwell Oy, Tampere, Finland
- With Host-Target add-on work with any compiler on any target
- With Bitcov add-on also on uController
- With Java and C# add-on also on JavaVM or DalvikVM

What does it

- measuring test coverage
- dynamic analyses
- at any testing phase
- for C or C++ code
- also Java and C#

Example report

```
Start/  End/
  True False - Line Source
-------------------------------------------------------------------------------
                    1 /* File calc.c ------------------------------------------------ */
                    2 #include "calc.h"
                    3 /* Tell if the argument is a prime (ret 1) or not (ret 0) */
Top
     9      0       4 int is_prime(unsigned val)
                    5 {
                    6     unsigned divisor;
                    7
     2      7       8     if (val == 1 || val == 2 || val == 3)
     1              8       T || _ || _
     0        -     8       F || T || _
     1              8       F || F || T
            7       8       F || F || F
     2              9         return 1;
     5      2      10     if (val % 2 == 0)
     5             11         return 0;
    58      2      12     for (divisor = 3; divisor < val / 2; divisor += 2)
                   13     {
     0     58  -    14         if (val % divisor == 0)
     0         -    15             return 0;
                   16     }
     2             17     return 1;
                   18 }
***TER 82% (14/17) of SOURCE FILE calc.c
```

CTC++ preparation

- Add arm-linux-androideabi-gcc to the ctc.ini settings
- Copy Host-Target source files to jni/
- Add targ*.c to jni/Android.mk

```
LOCAL_SRC_FILES := plasma.c targdata.c \
targcust.c targsend.c
```

- Add permission to write to sdcard (or use other arrangement)

```
<uses-permission
android:name=
"android.permission.WRITE_EXTERNAL_STORAGE"
 />
```

- Mention function name to dump data out

```
EMBED_FUNCTION_NAME = \
Java_com_example_plasma_PlasmaView_renderPlasma
```

Building NDK application with CTC++

Building

- Compiler with

```
ndk-build TARGET_CC= \
"ctc -i m arm-linux-androideabi-gcc"
```

- Build the package

```
ant debug
```

- Install with

```
adb install -r bin/Plasma-debug.apk
```

- Run the application

Data transfer and reporting

Getting report

- Getting data

```
adb pull /sdcard/MON.txt .
```

- Convert date

```
ctc2dat -i MON.txt
```

- **MON.dat** gets born
- proceed as on host:

```
ctcpost MON.dat -p - | ctc2html
```

- Have a look

```
firefox CTCHTML/index.html
```

Building Java application with CTC++

Compiling

- `ctc.jar` has to be added to the `CLASSPATH`
- CTC++ android library project stored somewhere
- Mention the function to trigger data write out in ctc-java-cs.ini

  ```
  EMBED_FUNCTION_NAME=onPause
  ```

- Build it

  ```
  ant debug \
  -Dbuild.compiler=fi.testwell.ant.ctc \
  -Dbuild.compiler.ctcopts="-i m"
  ```

- `adb install yourpackage.apk` on your Android
- Run it, play around, get coverage data

Data transfer until report

Getting report

- Getting data

```
adb pull /sdcard/MON.txt .
```

- Convert date

```
ctc2dat -i MON.txt
```

- **MON.dat** gets born
- proceed as on host:

```
ctcpost MON.dat -p - | ctc2html
```

- Have a look

```
firefox CTCHTML/index.html
```

Thank you

Thank you very much for attention

Availability

- Evalversions available at http://www.verifysoft.de/