Søren Hansen <sha@ase.au.dk>

Debugging on host and cross debugging on target

V1.1

Introduction

Debugging on host and cross debugging on target

Prerequisites

To complete this exercise you must:

- Have access to a target
- Have installed `ddd` and `valgrind`

Goal

Upon successful completion of this exercise, you will:

- Have gained insight into how to debug using `gdb`.
- How debugging is done using `ddd`.
- How you enable core dumps and how you use them to find errors in you code.
- Have gained insight into how cross debugging is done.
- Have gained insight into how `valgrind` and co. can aid the development process.

In the exercises below you will need an application that can be compiled for both *host* and *target*. The application in question *must* utilize threads and synchronization mechanisms.

Exercise 1 Debugging using gdb

Compile your application for the host, and single step through it using `gdb`.

- Insert break points.
- See the different threads.
- Get a backtrace of the different threads and acknowledge what they are doing.

Finally introduce a *segmentation fault* and proceed to find it using `gdb`

Exercise 2 Using ddd

New repeat Exercise 1 using `ddd`.

Exercise 3 Cross debugging

Recompile the test application for target. Setup cross debugging via gdb, repeat and find the error again.

Exercise 4 Core dumps

Enable *core-dumps* on target and run the error prone program once again. Use the dumped core to find the error in gdb on the host.

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Exercise 5 Cross debugging using ddd

New repeat Exercise 3 using `ddd`.

Exercise 6 Valgrind & Hellgrind

Finally try to run the test application on the host using `valgrind`. How many memory leaks are actually in the program? Are there any thread issues?

Introduce *at least* **3** thread issues and see how the output reflects these changes. Similary introduce *at least* **2** different memory problems.

Explain the chosen problems and why they are important to understand.

If you cannot imagine such a problem, how will ever find it?