



GDB Roadmap BoF

Jan Kratochvíl <jan.kratochvil@redhat.com>

July 9, 2012

Agenda

- Archer Roadmap
- C++Parser
- Fix-and-Continue
- GDB in C++
- C Plugins vs. Python Scripting
- Maintenance Issues
- Regression Tester
- Tcl/DejaGnu of the testsuite

Archer Roadmap

- Archer — draft patches hosting
 - `http://sourceware.org/gdb/wiki/ProjectArcher`
 - `git://sourceware.org/git/archer.git`
 - `http://sourceware.org/ml/archer/`
- real RFE list, sourceware.org BZ is still unsorted
- free read/write GIT repository (sourceware account needed)
- highlights
 - C++parser - GDB's own parser, GCC, LLVM
 - fix-and-continue
 - linux-nat replacement by gdbserver
 - ...

C++Parser

- GCC based by Keith Seitz: archer-expr-plugin
 - <http://sourceware.org/ml/archer/2011-q1/msg00122.html>
 - slow, up to 10 seconds
- GDB's own parser will never match perfectly
- always unsupported (unless LLVM JIT):
 - ({ embedded; statement; })
 - passing C++11 lambda functions by hand

Fix-and-Continue

- implemented for gdb-6.3 by Apple
 - Jason Molenda:
 - <http://sourceware.org/ml/gdb/2003-06/msg00531.html>
 - <http://opensource.apple.com/source/gdb/gdb-1752/>
- *nop* in the prologue to insert *jmp*
- new function as *-fPIC -shared*
- *dlopen-ed*
- issue of static symbols redirection (not using GOT)

GDB in C++

- benefits listed by Tom Tromey
 - <http://sourceware.org/ml/gdb/2012-04/msg00044.html>
 - some bugfixes needed due to GDB's own C++-in-C framework
- remains bootstrapping of new archs
 - <http://gcc.gnu.org/wiki/CppBuildStatus>

C Plugins vs. Python Scripting

- sometimes people ask for (non-existing) plugins API (=C)
 - they are given existing Python API
- GDB jit-reader exists (+reqd `plugin_is_GPL_compatible`)
- GCC has now fresh `gcc-python`, comparison?

Maintenance Issues

- patch tracking, technical reviewing (Pending Patches BoF)
 - Rietveld extension
- conversion to GIT
- regression tester

Regression Tester

- no public service (besides GCC compile farm hosts)
- GDB (and GCC) automated testing is post-commit
- security: public service or existing contributors?
- many combinations (DWARF2-4, stabs, archs etc.)
 - one run is 3 hours
- *hammock*: insecure, no non-mock compatible OSes

Tcl/DejaGnu of the testsuite

- difficult for new contributors, repeating mistakes
 - two kinds of escaping, anchoring,
 - out of command sync (missing `$gdb_prompt`), ...
- repeating mistakes even by long-term contributors
- compatibility with existing board files
- existing remote host and remote target setups
- it should not be based on the CLI interface

Project - VLA - Dynamic Types

= Variable Length Arrays: `char array[param + 1]`

- abstract types and value-bound types exist
- fully dynamic accessors or `check_typedef`-like constification?
- slow vs. need of two inferior type types (*struct type **).



The end.

Thanks.