

NGDB - new debugger for FreeBSD

Doug Rabson
dfr@freebsd.org

Why bother?

- GDB is hard to work with
- There must be a better way
- Writing new code is fun!

Goals

- A command-line debugger with a familiar interface (maybe add a GUI later)
- Support ELF, dwarf debug information
- Debug local programs with ptrace
- Postmortem debugging using core files
- GDB-remote protocol for kernel debugging

Goals

- Support for shared libraries
- Threaded programs, including ones that use TLS
- Debugging C, C++, D programs
- Support for multiple platform types
- Any other features of GDB that I actually use

Non goals

- Support for obsolete file formats
- Support for obsolete/obscure platforms
- Languages I don't use

How does it work?

- Simple set of classes and interfaces
- Tied up by a command line interface
- No reason a GUI interface couldn't use the same classes

How does it work?

Target

- Responsible for examining the debugged program
- Target has a list of threads and modules
- Sets/clears breakpoints
- Interacts via TargetListener, TargetBreakpointListener interface

How does it work?

TargetThread

- One of these for each active thread in the running program
- TargetListener is called on thread create, destroy
- Accessors for unique thread ID and to access thread state

How does it work?

TargetModule

- One of these for each ELF module - main program or shared library
- TargetListener called on shared library load, unload
- Access to low-level symbols
- If present, supplies a DebugInfo object for more detailed information

How does it work?

MachineState

- Processor state for a single thread
- Access register state
- Disassemble code

How does it work?

Debugger

- Command line interface
- Keeps track of the target, threads, modules
- Command objects implement all the features

How does it work?

DebugInfo

- Accesses dwarf debug information
- Unwinds stack frames using dwarf unwind information
- Describes how to access variables etc

How does it work?

Location

- Describes how to find a value
 - Register
 - Memory
 - Subrange of some other Location
 - Concatenation of some other Locations

How does it work

Language

- Responsible for displaying expressions, types etc.
- Parse expressions

Where can I get it?

- GIT Repository at <http://people.freebsd.org/~dfr/ngdb.git>
- Get a D compiler from <http://www.digitalmars.com/d/download.html>

Questions?