RTEMS/GDB User's Guide

Edition 4.6.0, for 4.6.0

30 August 2003

Emmanuel Raguet and Eric Valette, Canon Research Centre
On-Line Applications Research Corporation

On-Line Applications Research Corporation
TEXinfo 2002-11-25.11

The RTEMS Project is hosted at `http://www.rtems.com`. Any inquiries concerning RTEMS, its related support components, its documentation, or any custom services for RTEMS should be directed to the contacts listed on that site. A current list of RTEMS Support Providers is at `http://www.rtems.com/support.html`.

Table of Contents

1 Introduction

GDB 4.17 is a source-level symbolic debugger for RTEMS environment. This version is an extension of GNU GDB version 4.17, adapted for the debugging of RTEMS applications in a heterogeneous host-target environment.

This paper documents **only** the RTEMS/GDB extensions to GNU GDB 4.17, as well as its use in the RTEMS environment. The reader is assumed to be familiar with GNU GDB.

The RTEMS project would like to thank Emmanuel Raguet (raguet@crf.canon.fr) and Eric Valette (valette@crf.canon.fr) of the Canon Research Centre France S.A. for developing and documenting RTEMS/GDB and submitting it.

2 Software Architecture

RTEMS/GDB host-target debugger is architectured around:

- One debugger, running on the host.
- One debug server, running on the target.

The host debugger communicates with the debug server using Sun RPC over UDP/IP (Ethernet).

When opening a debugging session, the user takes the control over the entire target system. The target system is viewed as a single multi-threaded process.

The debugger can observe the state of the system objects, without stopping the entire system. However, breakpoints and stepping affect (stop) the entire system.

For a given target, only one single system debugging session is possible at a time.

3 Getting Started

This section describes the steps which are necessary to set up a debugging session with RTEMS/GDB.

3.1 Compiling The System Components

The components that need to be debugged have to be compiled with the -g option in order the generated file to contain the required information for the debug session.

3.2 Starting a debugging session

After having launched GDB, the user must connect to the RTEMS target with the following commands:

```
(gdb) setrpcmode sun
(gdb) setdaemontype rdbg
(gdb) target rtems target-name
```

Then he can use the file GDB command to specify the binary to be debugged.

This initialization phase can be written in a ''.gdbinit'' file. Each time, the user will launch GDB, it will execute this initialization sequence.

3.3 Attaching To The System

The entire target system is viewed as a single multi-threaded process, the identifier of which is 1. To attach, use:

```
(gdb) attach 1
```

When successfully attached, the user can control the execution of the target system from the debugger.

4 Commands

The usual GDB commands are available to display or modify data structures and machine registers, disassemble code, display source listings, set and reset breakpoints, control the debuggee execution, etc. Refer to Debugging with GDB for a complete documentation of the standard GDB commands.

GDB 4.17 has been extended with a number of commands specifically for multi-threaded debugging within a RTEMS host-target environment. These commands are described in this section.

4.1 Host-target connection control

- (gdb) setrpcmode sun : specifies the SUN RPC type.
- (gdb) setdaemontype rdbg : specifies the RDBG debug server type.
- (gdb) target rtems target-name : specifies the target type.

4.2 System status information

- (gdb) info threads : gives a list of all threads in the debuggee, their names, identifiers and states. It also indicates which thread is the target thread.

4.3 Thread control

- (gdb) thread target [id] : allows the user to get/set the target thread. The argument id is either the thread's symbolic name or its local identifier.
- (gdb) thread detach [id...] : all stops (breakpoints) of some specified thread(s) can be ignored using the thread detach command. Each time a detached thread hits a breakpoint, RTEMS/GDB will automatically restart the debuggee without passing control to the user.
- (gdb) thread attach [id...] : (re)attach specified thread(s).

5 Troubleshooting

This section describes symptoms in some situations where the debugger does not work correctly, and it gives guidelines for resolving the problems encountered.

5.1 Target machine name unknown on the host

RTEMS/GDB fails to connect to target machine named my_target:

```
(gdb) target rtems my_target

Attaching remote machine across net... \
      Invalid hostname. Couldn't find remote host address.
```

==> *Ask your system administrator to add an entry with the Internet number of machine my_target in the file /etc/hosts on your host machine.*

5.2 Debug server not present

The target rtems command fails:

```
(gdb) target rtems my_target

Attaching remote machine across net... RPC timed out. Couldn't connect
to remote target
```

==> *Verify that the target system is properly configured, and is running the debugging daemon and communication stack.*

6 Debug Session Example

The following is a debug session. Note that some lines have been broken to aviod formatting problems in the printed document.

```
GNU gdb 4.17
Copyright 1998 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and
you are welcome to change it and/or distribute copies of it under
certain conditions.  Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.
Type "show warranty" for details.
This GDB was configured as --host=i686-pc-linux-gnu --target=i386-rtems.
(gdb) setrpcmode sun
(gdb) setdaemontype rdbg
(gdb) target rtems usnet-test
Attaching remote machine across net...
Connected to usnet-test.
Now the "run" command will start a remote process.
(gdb) file /buildr4/pc386/tests/debug.exe
Reading symbols from /buildr4/pc386/tests/debug.exe...done.
(gdb) attach 1
Attaching program: /buildr4/pc386/tests/debug.exe pid 1
0x22fe65 in enterRdbg ()
(gdb) info threads
There are 8 threads:
Id.                Name       Detached   Suspended
134283273          Rini       No         No <= current target thread
0x22fe65 in enterRdbg ()
134283272          Evnt       No         No
_Thread_Dispatch () at /rtems4/cpukit/score/src/thread.c:315
134283271          SPE2       No         No
_Thread_Dispatch () at /rtems4/cpukit/score/src/thread.c:315
134283270          SPE1       No         No
_Thread_Handler () at /rtems4/cpukit/score/src/thread.c:1107
134283269          RDBG       No         No
0x22fe65 in enterRdbg ()
134283268          SCrx       No         No
_Thread_Dispatch () at /rtems4/cpukit/score/src/thread.c:315
134283267          SCtx       No         No
_Thread_Dispatch () at /rtems4/cpukit/score/src/thread.c:315
134283266          ntwk       No         No
_Thread_Dispatch () at /rtems4/cpukit/score/src/thread.c:315
(gdb) b init.c:92
Breakpoint 1 at 0x200180: file \
       /rtems4/c/src/tests/samples/debug/init.c, line 92.
(gdb) c
```

```
Continuing.
Thread 134283273 (Rini) has been deleted.
[Switching to Rtems thread 134283271 (Not suspended) \
    ( <= current target thread )]
Breakpoint 1, example2 (argument=4) at \
    /rtems4/c/src/tests/samples/debug/init.c:92
92 tuto += tuti;
(gdb) b init.c:66
Breakpoint 2 at 0x200128: file \
    /rtems4/c/src/tests/samples/debug/init.c, line 66.
(gdb) c
Continuing.
[Switching to Rtems thread 134283270 (Not suspended) \
    ( <= current target thread )]
Breakpoint 2, example1 (argument=4) at \
    /rtems4/c/src/tests/samples/debug/init.c:66
66 toto += titi;
(gdb) c
Continuing.
[Switching to Rtems thread 134283271 (Not suspended) \
    ( <= current target thread )]
Breakpoint 1, example2 (argument=4) at \
    /rtems4/c/src/tests/samples/debug/init.c:92
92 tuto += tuti;
(gdb) s
93 if (print_enable2)
(gdb) detach
Detaching program: /buildr4/pc386/tests/debug.exe pid 1
Warning: the next command will be done localy!
If you want to restart another remote program, reuse the target command
(gdb) quit
```