

The Embecosm Proxy GDB RSP Server

User Guide

Jeremy Bennett
Embecosm Limited
Issue 1 for rsp-proxy 1.1

This file documents the Embecosm Proxy GDB RSP Server, rsp-proxy.

Copyright © 2009 Embecosm Limited.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Published by Embecosm Limited

Table of Contents

Scope of this Document	1
1 About the Embecosm Proxy GDB RSP Server	2
2 Installation	3
2.1 Preparation	3
2.2 Configuring the Build	3
2.3 Building and Installing	3
2.4 Known Problems and Issues	3
3 Usage	4
3.1 Starting rsp-proxy	4
3.2 Terminating rsp-proxy	4
3.3 Unsupported GDB Commands	4
4 Configuration	5
4.1 General Format	5
4.2 Specifying Machine Properties	5
4.3 Specifying Registers	5
4.4 Specifying Memory Blocks	6
4.4.1 Specifying Memory in Bytes	6
4.4.2 Specifying Memory in Words	6
4.5 Formal Syntax	7
5 Example	8
5.1 Overview	8
5.2 Example Configuration File	9
5.3 Typical Output	11
6 GNU Free Documentation License	14
Index	21

Scope of this Document

This document is the user guide for rsp-proxy, the Embecosm Proxy GDB RSP Server.

1 About the Embecosm Proxy GDB RSP Server

rsp-proxy provides a server interface to allow GNU Debugger clients for bare metal remote targets (typical of the embedded world) to be tested before the actual target is available.

Typically such targets are provided as models or interfaces to real silicon, and availability of a working target is the limiting step on delivery. The rsp-proxy allows the maximum testing to be carried out before the target is available, thus reducing time to delivery.

rsp-proxy models the target as a bank of registers and one or more banks of (non-overlapping) memory. This is initialized from a configuration file and then accessed and modified through any RSP packets which access registers or memory.

If the user requests, full tracing of packets can be enabled.

2 Installation

Installation follows standard GNU protocols.

2.1 Preparation

Unpack the software and create a *separate* directory in which to build it:

```
tar jxf embecosm-esp7-rsp-proxy-1.1.tar.bz2
mkdir build-rsp-proxy
cd build-rsp-proxy
```

2.2 Configuring the Build

Configure the software using the `configure` script in the main directory.

There several options available, all of which are standard to GNU `configure` scripts. Use `configure --help` to see all the options. The most useful is `--prefix` to specify a directory for installation of the tools.

2.3 Building and Installing

The tool is then built with:

```
make all
make install
```

The main installation directory should have been specified using `--prefix` at configuration time. These commands will install the `rsp-proxy` tool in the `'bin'` sub-directory, the example in the `'share'` sub-directory and this documentation in `info` format in the `'share/info'` sub-directory.

Note: Testing `rsp-proxy` with `make check` is not yet supported.

The documentation may be created and installed in alternative formats (PDF, Postscript, DVI, HTML) with for example:

```
make pdf
make install-pdf
```

2.4 Known Problems and Issues

The following problems and issues are known about with `rsp-proxy` 1.1. Notify other issues by email to <mailto:support@embecosm.com>.

- There is no check that all the registers have been specified.
- A word is assumed to be 4 bytes

3 Usage

3.1 Starting `rsp-proxy`

The general form of the command is:

```
rsp-proxy config-file port [trace]
```

The *config-file* is a file specifying the initial configuration of `rsp-proxy`. The port is the TCP/IP port on which the server should listen. If the optional final `trace` argument is specified, the server will report on each RSP packet it sends and receives.

The program acts as a RSP server, and will respond to RSP packets from a GDB client, using the information from the configuration file.

3.2 Terminating `rsp-proxy`

There is no clean way of terminating `rsp-proxy`. Just kill it with `ctrl-C`.

3.3 Unsupported GDB Commands

The only GDB commands which are supported are those which rely on accessing registers and memory. `rsp-proxy` will attempt to give meaningful response to other packets as far as is possible. For example `step` or `continue` packets will respond as though the program has immediately halted at a breakpoint without the program counter changing.

This means that `stepi` and `continue` commands work fine, but `step` and `next` will hang, as the GDB client keeps stepping the target, looking for the program counter to advance.

It is possible for clients to detach from the target (using the `detach` command) and then reattach using a new `target remote` command.

4 Configuration

The configuration file describes an initial state of the CPU registers and memory

4.1 General Format

The configuration file consists of three sections.

- A description of the machine properties, introduced by the keyword `NAME`.
- A description of the registers and their initial state, introduced by the keyword `REGISTERS`.
- A description of a memory block and its initial values, introduced by the keywords `BYTE MEMORY` or `WORD MEMORY`. This block may occur several times to describe multiple blocks, such as program memory, stack memory and static data.

The configuration is in free format. Comments may be introduced, C++ style, by `//`.

Note: The C style quotation `/* ... */` is *not* supported

4.2 Specifying Machine Properties

The machine properties section must appear first. The keyword, `NAME`, is followed by a parenthesized, comma separate list containing a textual name for this configuration (in double quotation marks) and the endianness of the target specified by the keywords `BIG ENDIAN` or `LITTLE ENDIAN`.

For example the configuration of a big endian OpenRISC 1000 target could be specified as:

```
NAME ( "OpenRISC 1000", BIG ENDIAN)
```

4.3 Specifying Registers

The registers are specified immediately after the machine properties. The section is introduced by the keyword `REGISTERS` followed by the number of registers in parenthesis.

This number should match the number of *real* registers specified in the GDB client's architecture definition. I.e not including any pseudo-registers. For example in the OpenRISC 1000 architecture, the real registers that GDB knows about are the 32 general purpose registers, the Previous Program Counter, the Next Program Counter and the Supervision Register, so its register specification starts as follows.

```
REGISTERS (35)
```

This declaration is followed by a comma separated triple in braces (`{` and `}`) for each register. The triple specifies a name of the register (in double quotes), its width in bits and its initial value. The registers must be specified in the same order that they are specified in the GDB client architecture.

The following example specifies the OpenRISC 1000 registers for a program about the execute the instruction at location `0x1030`, with a stack pointer at `0x7eb4` and frame pointer at `0x7ebc`, and return address (in the link register) of `0x12b0`.

```
REGISTERS ( 35 )
{ "gpr0", 32, 0 }           // Always zero
{ "gpr1", 32, 0x7eb4 }     // Stack pointer
{ "gpr2", 32, 0x7ebc }     // Frame pointer
{ "gpr3", 32, 0 }         // Start of argument registers
{ "gpr4", 32, 0 }
{ "gpr5", 32, 0 }
{ "gpr6", 32, 0 }
{ "gpr7", 32, 0 }
{ "gpr8", 32, 0 }         // End of argument registers
```



```

{ "gpr9", 32, 0x12b0 }           // Link register
{ "gpr10", 32, 0 }

... More general purpose registers initialized to zero ...

{ "gpr31", 32, 0 }
{ "ppc", 32, 0x102c }           // Previous program counter
{ "npc", 32, 0x1030 }           // Next program counter
{ "sr", 32, 0 }                 // Supervision register

```

The bit size and initial value may be specified in octal, decimal or hexadecimal using the conventional C/C++ notation.

4.4 Specifying Memory Blocks

The specification of memory blocks follows the machine description and register specification. Multiple blocks may be specified.

Memory is assumed to be byte addressed. It may be initialized as either bytes or words (in which case the endianness is taken into account).

4.4.1 Specifying Memory in Bytes

Byte memory is specified by the keywords `BYTE MEMORY` followed by a comma separated list within parentheses specifying the start address of the memory and its size in bytes.

For example a 256 byte memory at location 0x8000 would be specified as follows.

```

BYTE MEMORY (0x8000, 256)

```

This declaration is optionally followed by a comma separated list of bytes giving initial values of locations. Not all (or any) locations need be specified. Any unspecified will be uninitialized. There may not be more bytes specified than the size of the memory. Thus the first 12 bytes of the memory specified at location 0x8000 could be set to the ASCII representation of C string "Hello World" as follows.

```

BYTE MEMORY (0x8000, 256)
0x48, 0x65, 0x6c, 0x6c, 0x6f,
0x20,
0x57, 0x6f, 0x72, 0x6c, 0x64,
0x00

```

The byte list ends at the end of file, or the start of the next memory block specification.

4.4.2 Specifying Memory in Words

Word memory is specified by the keywords `WORD MEMORY` followed by a comma separated list within parentheses specifying the start address of the memory and its size in bytes.

Note: The size is still in *bytes*, even though this is word memory.

For example a 0x3a0 byte long memory at location 0x1000 would be specified as follows.

```

WORD MEMORY (0x1000, 0x3a0)

```

This declamation is optionally followed by a comma separated list of words giving initial values of locations. These will be stored in bytes of memory according to the endianness specified in the machine description.

Note: Words are assumed to be 4 bytes long. This is a limitation of the current implementation.

As with byte memory, not all (or any) locations need be specified. Any unspecified will be uninitialized. There may not be more words specified than the size of the memory. The memory

specified at location 0x1000 could be initialized with an OpenRISC 1000 program image as follows:

```
WORD MEMORY (0x1000, 0x3a0)
0x9c21fff8, 0xd4011000, 0x9c410008, 0xd7e21ffc,
0x15000001, 0x84410000, 0x44004800, 0x9c210008,
0x9c21fff8, 0xd4011000, 0x9c410008, 0xd7e21ffc,
0x15000004, 0x84410000, 0x44004800, 0x9c210008,

... lots more similar ...

0x07ffffb0, 0x15000000, 0x9c60002a, 0x07ffff74,
0x15000000, 0x18600000, 0xa86313bd, 0x07ffffa9,
0x15000000, 0x07ffffe3, 0x15000000, 0x8562fffc,
0x85210000, 0x84410004, 0x44004800, 0x9c21000c
```

The word list ends at the end of file, or the start of the next memory block specification.

4.5 Formal Syntax

The syntax of the configuration file in Backus-Naur Form (BNF) is:

```
description      -> nameClause registerClause memoryClauseList
nameClause      -> NAME ( nameParams )
nameParams      -> string , endianness
endianness      -> LITTLE ENDIAN | BIG ENDIAN
registerClause  -> REGISTERS ( number ) registerList
registerList    -> register | register registerList
register        -> { string , number , number }
memoryClauseList -> memoryClause | memoryClause memoryClauseList
memoryClause   -> byteMemoryClause | wordMemoryClause
byteMemoryClause -> BYTE MEMORY ( memoryParams ) memoryValues
wordMemoryClause -> WORD MEMORY ( memoryParams ) memoryValues
memoryParams    -> number , number
memoryValues   -> number | number , memoryValues
```

Comments are introduced by // and last to the end of line C++ style.

5 Example

The `example` directory within the distribution contains an example configuration file, for a big-endian OpenRISC 1000. It represents the target during the execution of a simple test program.

5.1 Overview

The main source code file, `hello.c` defines a program to print out some strings and perform a simple calculation.

```
#include "utils.h"

void level2() {
    simexit( 42 );
}

void level1() {
    level2();
}

main()
{
    simputs ("Hello World!\n");
    simputs ("The answer is ");
    simputn (6 * 7);
    simputs ("\n");
    level1();
}
```

This program runs under the control of a simple bootloader, stored at the reset vector location, 0x100.

```
.org    0x100          # The reset routine goes at 0x100

.global _start
_start:
    l.addi  r1,r0,0x7f00  # Set SP to value 0x7f00
    l.addi  r2,r1,0x0     # FP and SP are the same
    l.mfspr r3,r0,17     # Get SR value
    l.ori   r3,r3,0x2    # Set TT exception enable bit
    l.jal   _main        # Jump to main routine
    l.mtspr r0,r3,17     # Enable exceptions (DELAY SLOT)
```

The bootloader establishes a stack (falling for the OpenRISC 1000) at location 0x7f00.

Simple utility functions provide character, string and number output to a monitor. One of these is `simputs`.

```
void simputs( char *str )
{
    int i;

    for( i = 0; str[i] != '\0' ; i++ )
    {
        simputc( (int)(str[i]) );
    }
} /* simputs() */
```

The code can be compiled using the OpenRISC 1000 GNU tool chain. However for convenience a compiled image, 'hello', is provided with the distribution.

The code is loaded at address 0x1000. In this example, the code for `main` starts at 0x1338, for `simputs` at 0x113c and for `simputc` at 0x1020.

5.2 Example Configuration File

The example configuration file is in the file 'or1k.cfg'. It represents the state of the target at the start of executing the code of `simputc`, called from `simputs`, following its first call from `main`.

The stack frame is for `simputc`, with stack pointer 0x7eb4 and frame pointer 0x7ebc. The next instruction to be executed is at location 0x1030 (immediately after the `simputc` function prologue). The return address, held in the link register, `gpr9`, is 0x12b0, within the code of `main`.

The machine description specifies this is an OpenRISC 1000 and big endian.

```
NAME ( "OpenRISC 1000", BIG ENDIAN)
```

The register description specifies the 32 General Purpose Registers, the Next and Previous Program Counters and the Supervision Register. These are initialized to correspond to the program and stack described earlier.

```
REGISTERS ( 35 )
{ "gpr0", 32, 0 }           // Always zero
{ "gpr1", 32, 0x7eb4 }     // Stack pointer
{ "gpr2", 32, 0x7ebc }     // Frame pointer
{ "gpr3", 32, 0 }         // Start of argument registers
{ "gpr4", 32, 0 }
{ "gpr5", 32, 0 }
{ "gpr6", 32, 0 }
{ "gpr7", 32, 0 }
{ "gpr8", 32, 0 }         // End of argument registers
{ "gpr9", 32, 0x12b0 }    // Link register
{ "gpr10", 32, 0 }

... More general purpose registers initialized to zero ...

{ "gpr31", 32, 0 }
{ "ppc", 32, 0x102c }     // Previous program counter
{ "npc", 32, 0x1030 }    // Next program counter
{ "sr", 32, 0 }         // Supervision register
```

There are four memory blocks.

- A block to specify the interrupt vectors, particularly the bootloader and the reset vector.
- A block to specify the main body of the code to be loaded at address 0x1000.
- A block to specify the static data (strings) used by the program at address 0x13a0.
- A block to specify the current values on the stack frame growing down from 0x7f00.

The interrupt vector block specifies 256 bytes of zeros, followed by the interrupt vector code. The remainder of the interrupt vectors can be uninitialized, since we will not be analyzing them with GDB.

```
WORD MEMORY (0x0, 0x1000)
0x00000000, 0x00000000, 0x00000000, 0x00000000,
0x00000000, 0x00000000, 0x00000000, 0x00000000,
```

```

... 60 more similar lines ...

0x00000000, 0x00000000, 0x00000000, 0x00000000,
0x00000000, 0x00000000, 0x00000000, 0x00000000,
0x9c207f00,                               // l.addi  r1,r0,0x7f00
0x9c410000,                               // l.addi  r2,r1,0
0xb4600011,                               // l.mfspr r3,r0,0x11
0xa8630002,                               // l.ori   r3,r3,0x2
0x0400048a,                               // l.jal   <main>
0xc0001811                               // l.mtspr r0,r3,0x11

```

The main program image is a straight word dump in hex of the program image.

```

WORD MEMORY (0x1000, 0x3a0)
0x9c21fff8, 0xd4011000, 0x9c410008, 0xd7e21ffc,
0x15000001, 0x84410000, 0x44004800, 0x9c210008,
0x9c21fff8, 0xd4011000, 0x9c410008, 0xd7e21ffc,
0x15000004, 0x84410000, 0x44004800, 0x9c210008,

```

... lots more similar ...

```

0x07ffffb0, 0x15000000, 0x9c60002a, 0x07ffff74,
0x15000000, 0x18600000, 0xa86313bd, 0x07ffffa9,
0x15000000, 0x07ffffe3, 0x15000000, 0x8562fffc,
0x85210000, 0x84410004, 0x44004800, 0x9c21000c

```

The static data contains 3 strings used by the main program, “Hello World!\n”, “The answer is “ and “\n”.

```

// Static data
BYTE MEMORY (0x13a0, 31)
0x48, // 'H'
0x65, // 'e'
0x6c, // 'l'
0x6c, // 'l'

```

... 23 more characters ...

```

0x20, // ' '
0x00, // EOS
0x0a, // '\n'
0x00 // EOS

```

The stack frame shows the top 80 words of the stack, which is sufficient to include the stack frames of `main`, `simputs` and `simputc`. Any values expected to have been stored on the frame, including frame pointer links and return addresses must be initialized correctly.

```

WORD MEMORY (0x7eb0, 80)
0x00000000,
0x00007ef4, // Current SP, holds pointer to prev (simputs) FP
0x00000048, // Start of simputc frame, arg to simputc (ASCII 'H')
0x00001358, // End of simputs frame, holds return address (in main)
0x00007f00, // Holds pointer to prev (main) FP
0x00000000,
0x00000000,

```

```

0x00000048,          // Argument to pass to simputc (ASCII 'H')
0x00000000,
0x000013a0,          // Temporary value of str
0x000013a0,          // Temporary value of &str[i]
0x00000048,          // str[i]
0x00000000,
0x000013a0,          // Value of str
0x000013a0,          // Value of &str[i]
0x00000000,          // Value of i
0x000013a0,          // Start of simputs frame, arg to simputs (str)
0x00000118,          // End of main frame, holds return address (in start)
0x00007f00,          // Start of main frame, holds pointer to prev (start) FP
0x00000000

```

5.3 Typical Output

In one window, change to the example directory and start `rsp-proxy` with the example configuration file `'or1k.cfg'` on port 51000 with tracing enabled.

```
rsp-proxy or1k.cfg 51000 trace
```

`rsp-proxy` reports it is listening:

```
Listening for RSP on port 51000
```

In a second window, change to the example directory and start OpenRISC 1000 GDB. Load the symbols from the `'hello'` executable

```

or32-uclinux-gdb
Building automata... done, num uncovered: 0/216.
Parsing operands data... done.
GNU gdb 6.8
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-linux-gnu --target=or32-uclinux".
(gdb) file hello
Reading symbols from rsp-proxy/example/hello...done.
(gdb)

```

Now connect to `rsp-proxy` using the GDB `target remote` command.

```

(gdb) target remote :51000
Remote debugging using :51000
simputc (c=72) at utils.c:61
61      __asm__ __volatile__ ( "\tl.nop\t%0" : : "K"( NOP_PUTC ));
(gdb)

```

In the window running `rsp-proxy`, a trace of the RSP packets exchanged is shown, since `rsp-proxy` was started with the `trace` option:

```

RSP trace: getPkt: RSP packet: 10 chars, "qSupported"
RSP trace: putPkt: RSP packet: 14 chars, "PacketSize=119"
RSP trace: getPkt: RSP packet: 1 chars, "?"
RSP trace: putPkt: RSP packet: 3 chars, "S05"
RSP trace: getPkt: RSP packet: 4 chars, "Hc-1"
RSP trace: putPkt: RSP packet: 2 chars, "OK"

```



```
(gdb) info registers gpr1 gpr2 gpr9 ppc npc
gpr1          0x7eb4  0x7eb4
gpr2          0x7ebc  0x7ebc
gpr9          0x12b0  4784
ppc           0x102c  0x102c <simputc+12>
npc           0x1030  0x1030 <simputc+16>
(gdb)
```

The code can be disassembled, to check that the image specified in the configuration is as expected.

```
(gdb) disas simputc
Dump of assembler code for function simputc:
0x00001020 <simputc+0>: l.addi    r1,r1,-8
0x00001024 <simputc+4>: l.sw     0(r1),r2
0x00001028 <simputc+8>: l.addi    r2,r1,8
0x0000102c <simputc+12>:      l.sw     -4(r2),r3
0x00001030 <simputc+16>:      l.nop    4
0x00001034 <simputc+20>:      l.lwz   r2,0(r1)
0x00001038 <simputc+24>:      l.jr    r9
0x0000103c <simputc+28>:      l.addi   r1,r1,8
End of assembler dump.
(gdb)
```

All register and memory locations are writable. For example we could set the argument to `simputc`, `c`, held at offset `-4` on the current frame, to a different value.

```
(gdb) set *((int *)$fp - 1) = 'X'
(gdb) backtrace
#0  simputc (c=88) at utils.c:61
#1  0x000012b0 in simputs (str=0x13a0 "Hello World!\n") at utils.c:109
#2  0x00001358 in main () at hello.c:44
#3  0x00000118 in _start ()
(gdb) print (char)c
$2 = 88 'X'
(gdb)
```


6 GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled

“Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified

version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C) year your name.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover  
Texts. A copy of the license is included in the section entitled ‘‘GNU  
Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with  
the Front-Cover Texts being list, and with the Back-Cover Texts  
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

B

BNF syntax	7
bugs	3

C

configuring rsp-proxy	5
-----------------------------	---

E

example, configuration	9
example, source code	8
example, typical output	11

G

GDB <code>continue</code> command	4
GDB <code>detach</code> command	4
GDB <code>next</code> command	4
GDB <code>step</code> command	4
GDB <code>stepi</code> command	4

I

installing rsp-proxy	3
----------------------------	---

K

known issues	3
--------------------	---

L

license for rsp-proxy	14
-----------------------------	----

M

machine properties	5
memory specification	6
memory, byte size	6
memory, word size	6

P

problems	3
----------------	---

R

register specification	5
register, number of	5
register, size	5
running rsp-proxy	4

S

starting rsp-proxy	4
--------------------------	---

T

terminating rsp-proxy	4
-----------------------------	---

U

unsupported GDB commands	4
--------------------------------	---