



Addison-Wesley Professional Ruby Series

Troubleshooting Ruby Processes


Leveraging System Tools When the
Usual Ruby Tricks Stop Working

Philippe Hanrigou

 Addison-Wesley
Pearson Education

www.informIT.com/Ruby

What This Shortcut Covers.....	3
Ruby Troubleshooting:	
The Usual Suspects	4
raise, puts and debug()	4
irb or script/console	5
Writing a Test.....	5
Log Files	6
Sending Signals	7
Debugger and breakpoint	8
Sometimes You Need a Different Kind of Trick.....	8
lsuf	9
What is It?	9
Usage	11
Concrete Examples Using lsuf to Troubleshoot a Problem with a Ruby Process.....	16
Exploring Other Tricks	19
Exploring Other lsuf Options	20
What is lsuf Good For?	22
strace.....	22
What is It?	22
Interpreting strace Output.....	23
How Do I Find Out About a Specific System Call?.....	24
What is strace Good For?.....	26
Some Concrete Examples	27
Permission Denied	33
Other Interesting strace Options	33
strace Siblings.....	35
gdb	39
The Basics: Attaching to a Running Process and Getting the C-Level Backtrace.....	40
Raising a Ruby Exception from gdb to Get the Ruby Stack Trace.....	41
Easy Access to your gdb Tricks: Define Macros in .gdbinit	44
Pushing the Envelope: Evaluate Arbitrary Ruby Code from gdb	45
Learning about Ruby Interpreter Internals	50
What is gdb Good For?	51
A Classic Gotcha: Unattachable Processes	52
Conclusion	53
Endnotes	54
Acknowledgments.....	55
About the Author.....	56



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this work, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this work, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Visit us on the Web: www.awprofessional.com

Copyright © 2008 Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
United States of America
Fax (617) 671 3447

This product is offered as an Adobe Reader™ PDF file and does not include digital rights management (DRM) software. While you can copy this material to your computer, you are not allowed to share this file with others.

ISBN 0-321-54468-4

ISBN13 978-0-321-54468-1

Second release, December 2007

Troubleshooting Ruby Processes: Leveraging System Tools when the Usual Ruby Tricks Stop Working (Digital Short Cut)

Table of Contents

What This Shortcut Covers

Ruby Troubleshooting: The Usual Suspects

- raise, puts and debug()

- irb or script/console

- Writing a Test

- Log Files

- Sending Signals

- Debugger and breakpointer

- Sometimes You Need a Different Kind of Trick

Isof

- What is It?

- Usage

- Concrete Examples Using Isof to Troubleshoot a Problem with a Ruby Process

- Exploring Other Tricks

- Exploring Other Isof Options

- What is Isof Good For?

Table of Contents

strace

- What is It?

- Interpreting strace Output

- How Do I Find Out About a Specific System Call?

- What is strace Good For?

- Some Concrete Examples

- Permission Denied

- Other Interesting strace Options

- strace Siblings

gdb

- The Basics: Attaching to a Running Process and Getting the C-Level Backtrace

- Raising a Ruby Exception from gdb to Get the Ruby Stack Trace

- Easy Access to your gdb Tricks: Define Macros in .gdbinit

- Pushing the Envelope: Evaluate Arbitrary Ruby Code from gdb

- Learning about Ruby Interpreter Internals

- What is gdb Good For?

- A Classic Gotcha: Unattachable Processes

Conclusion

Endnotes

Acknowledgments

About the Author

Table of Contents