GDB Pocket Reference

Debugging goes with programming like thorns go with roses: you can’t have one without the other. The professional gardener uses only the best tools for gardening; the professional programmer uses only the best tools for programming and debugging.

In the open source and free software worlds, GDB, or the GNU Debugger, is the standard debugger. It ships with all BSD and GNU/Linux systems, and compiles and works “out of the box” on just about every other kind of commercial Unix system. (It even works on Microsoft Windows systems!)

Upon its release, GDB was quickly acknowledged as the best of its class, providing features and capabilities that are still lacking in other line-oriented debuggers. GDB has continued to evolve, keeping up with new hardware, operating system, and programming language features.

GDB is also the foundation for more than one open source graphical debugger. Even if you don’t use it directly, it’s there under the hood, waiting for you to unleash its full power.

Despite GDB’s popularity, until now there hasn’t been a brief guide or a quick reference to its many features. The *GDB Pocket Reference* fills this gap. It’s designed so that you can look up features quickly and effectively. If you aren’t a fluent user of GDB, the *GDB Pocket Reference* will show you what this debugger is capable of doing. If you are fluent, this book will keep the information you need right at your fingertips—without cluttering up either your physical or virtual desktop.

www.oreilly.com

Table of Contents

Introduction  1
Conventions Used in This Book  2
Conceptual Overview  3
Command-Line Syntax  7
Initialization Files  10
GDB Expressions  11
The GDB Text User Interface  14
Group Listing of GDB Commands  14
Summary of set and show Commands  21
Summary of the info Command  35
Alphabetical Summary of GDB Commands  37
Index  63
Introduction

The GNU Debugger, GDB, is the standard debugger on GNU/Linux and BSD systems and can be used on just about any Unix system with a C compiler and at least one of several well-known object file formats. It can also be used on other kinds of systems as well. GDB has a very rich feature set, making it the preferred debugger of many developers the world over.

This pocket reference provides a complete summary of GDB command-line syntax, initialization files, expressions, variables, and commands. It also describes the source code locations for GDB and two other graphical debuggers based on GDB.

A full introduction to GDB may be found in its documentation, which is included in the source code. This documentation is also available from the Free Software Foundation in Debugging with GDB: The GNU Source-Level Debugger, by Richard M. Stallman, Roland Pesch, Stan Shebs, et al.
Conventions Used in This Book

This book follows the typographic conventions that are outlined below:

**Constant width**
Used for directory names, commands, program names, functions, variables, and options. All terms shown in constant width are typed literally. It is also used to show the contents of files or the output from commands.

**Constant width italic**
Used in syntax and command summaries to show generic text; these should be replaced with user-supplied values.

**Constant width bold**
Used in examples to show text that should be typed literally by the user.

**Italic**
Used to show generic arguments and options; these should be replaced with user-supplied values. Italic is also used to indicate URLs, macro package names, filenames, comments in examples, and the first mention of terms.

$  
Used in some examples as the Bash, Bourne or Korn shell prompt.

*program*(N)*  
Indicates the “manpage” for program in section N of the online manual. For example, echo(1) means the entry for the echo command.

[]  
Surround optional elements in a description of syntax. (The brackets should never be typed.) Note that many commands show the argument [files]. If a filename is omitted, standard input (usually the keyboard) is assumed. End keyboard input with an end-of-file character.
Index | 69

winheight command, 20, 80
--write command-line option, 9
write parameter (set/show commands), 35

x command, 16, 21, 60
-x command-line option, 9