Build all open source software—including Linux, GNOME, KDE, StarOffice, and the Apache Web Server—and the applications that run on them

The Complete Reference

GCC

Develop and deploy software on any UNIX platform—including Linux and BSD

Install and use your own compiler for C, C++, Objective C, Fortran, Java, and Ada

Generate native executable code for dozens of different platforms

Arthur Griffith
Author, computer consultant

GCC: The Complete Reference

For Mary

Want to learn more?

We hope you enjoy this McGraw-Hill eBook! If you'd like more information about this book, its author, or related books and websites, please **click here.**

McGraw-Hill/Osborne
A Division of The McGraw-Hill Companies

0-07-222405-3

DOI: 10.1036/0072228164

GCC: The Complete Reference

Arthur Griffith

McGraw-Hill/Osborne
New York Chicago San Francisco
Lisbon London Madrid Mexico City
Milan New Delhi San Juan
Seoul Singapore Sydney Toronto

About the Author

Arthur Griffith has been involved with the development of compilers, interpreters, linkers, and assemblers since his first programming job in 1977, where he worked as a team member developing an assembler and linker for special-purpose computers. He then joined the maintenance group for a compiler of the PL/EXUS language, which had an underlying structure very similar to GCC. The next project was to write an interactive interpreter and compiler for a language named SATS.

The projects that followed these included the development of a Forth interpreter, extensions to a COBOL compiler, and the development of some special-purpose interpretive languages for machine control. One of these was an interactive command language providing multistation ground-based control of industrial satellite communications systems.

For the past few years, Arthur Griffith has turned to writing computer books, teaching programming online, and developing some software in Java. The programming books he has written range from *Java, XML, and Jaxp* to *COBOL for Dummies*. He has used GCC for many software-development projects, and with the inclusion of Java as one of the GCC languages, writing this book became his project of choice.

Contents at a Glance

Contents

Part III

Peripherals and Internals

Acknowledgments

I must thank Wendy Rinaldi at McGraw-Hill/Osborne for giving me the opportunity to write this book, and for her patience in the early days when it looked like it was going to take forever.

I want to thank Katie Conley for keeping me on track and heading in the right direction. She has a unique ability for keeping track of the status of the various parts of the book as it moves through the editing process. Bart Reed and I have a completely different take on the English language—his is both readable and correct. I want to thank Paul Garland for checking the technical accuracy of the book and pointing out the places where my imagination overtook the facts.

I must thank Margot Maley at Waterside for keeping my feet on the ground and my hands on the keyboard.

My understanding of how compilers work was a necessity for writing this book. I want to thank Dave Rogers for introducing me to the C language many years ago, and for drafting me to write a compiler for it. I also need to thank Ron Souder and Travis Mitchell for throwing me into some very strange projects that caused me to become immersed in some of the more obscure nooks and crannies of language processing and object code generation.

Perhaps most of all, I owe a great deal of thanks to the late Fred Lewis for introducing me to the fascinating world of compilers, assemblers, and linkers.