

PREFACE

Compilers are utilities that transform programs written in higher level languages like Pascal, C, and C++ into lower level languages like the assembly language program or the machine code. The machine code can be directly executed on a computer to perform various tasks. The assembly language program can be converted to machine code by using another utility called assembler and then executed on a computer.

This book describes the internals of a compiler detailing the steps used by it to transform a higher level language program into a lower language program.

Purpose

In my 18 years of industrial experience, I have had the opportunity to absorb a number of fresh computer science graduates in my team. My observation has been that the fresh computer science graduates get a theoretical level of understanding of the compilers and its functions during their course work. These young men and women often find it difficult to get into job assignments that deal with compilers and the associated tools. They usually have to be trained to make them ready for the job assignments involving compilers. One of the reasons for this gap is lack of textbooks that emphasise on practical application of principles of compilers like Syntax Analysis, Semantic Analysis, etc. This book bridges the gap and provides students of computer science good reading material to understand the basics of compilers.

During my survey of textbooks on compilers, I found that there was a dearth of textbooks that had the right mix of theory and practice for compiler construction. I did not find many textbooks in the market that focused on introducing compilers to the entry-level student. Most of the available textbooks were suitable for an advanced course in compiler construction rather than the introductory course. Most of the existing textbooks also had a heavy theoretical emphasis. I embarked on writing this book because I felt a genuine need for a textbook that had (a) focus on the basics of compilers (b) accent on the practical aspect and (c) suitability to an entry-level student.

Background

It was the summer of 2001 that I was contemplating on writing a book in the field of computer science as my service to the industry that gave me bread and butter. I was more inclined towards writing a book on the subject of Artificial Intelligence because (a) I had implemented many of the AI algorithms while competing at the Programmer of the Month (POTM) contest, during my stay in AT&T Paradyne at NJ, USA. (b) I was reasonably convinced that students would find my book encouraging, since they would see “Source Code” and not just Theory.

On 18 May 2001, I spoke to Prof. Khodanpur, Retired Head of the department of computer science at RV College of Engineering, Bangalore, on the subject of writing a book on Artificial Intelligence. He encouraged me to undertake writing a book on Principles of Compiler Design instead of Artificial Intelligence. Prof. Khodanpur impressed upon me that the students were struggling to get a grasp on the basics of compilers and it would do a world of good, if I can contribute something to help the students. That was the genesis of this book on the principles of compilers. Prof. Khodanpur subsequently reviewed my first output and gave me some suggestions for improvement. I have tried my best to keep up to his philosophy of ‘keeping it simple and correct’ throughout this book.

For 8 years, I have been relentlessly working on various aspects of this book. The Almighty chose me and bestowed upon me the strength and the means to carry on this incredible journey. The blessings of my parents helped me overcome many a challenge during the process of making this book. The inspiration, happiness and purpose came from my constant companion and buddy—Pranav. There were numerous times in this span of 8 years, where I felt that I cannot go any further and thought of giving it all up. Fortunately, God willed it otherwise and today I live to see my dream fulfilled. It has been a fantastic experience writing this book and I hope that you find reading the book equally pleasurable.

Target Readers

The textbook is intended for an introductory level student who is familiar with C/C++ Programming. The goal is to cover the basics of the compiler theory rather than being an exhaustive complete reference in the subject. The textbook would be ideal for graduate/undergraduate level of students taking an introductory course on the compilers. It is particularly well suited for undergraduate students of computer science and information technology engineering. This book can also be used by professionals who wish to understand the basics of compilers in order to prepare themselves for working on projects based on compilers.

Salient Features

The salient features of the book are

- Simple chapter organisation based on different stages of a compiler
- Easy narrative style of explanation with emphasis on basic principles
- Numerous examples and illustrations clarifying concepts
- Incremental development of a Toy C language compiler
- References to the behaviour of production compilers

Organisation

The textbook is organised into 7 chapters. The first chapter provides an introduction to compilers. It talks about the different utilities that participate in the compilation process. It details the functionality of a compiler, describing briefly each stage of the compilation. This forms the basis for the rest of the book.

The different stages of compilation are described in successive chapters starting from Chapter 2. Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code Generation and Target Code generation are the topics for discussions in Chapters 2,3,4,5 and 6 respectively.

The techniques for optimising the intermediate code and the target code are described in Chapter 7—Optimisation.

The approach taken in each of the chapters is to introduce the theory with suitable practical examples. Algorithm description is usually followed by an implementation and demonstrated by an example. A number of examples use the C language compiler of GNU's compiler collection (gcc) as a reference to illustrate the behaviour of the compilers.

Online Learning Centre

The book needs to be read along with source code for examples that can be downloaded from <http://www.mhhe.com/raghavan/pcd>. These examples have been compiled and tested on CYGWIN 1.5 platform on my home x86 PC running Win98. I have also checked the examples on LINUX platform. I have used gcc 3.4, flex 2.5 and bison 2.3 for compiling and testing the examples. An older version of gcc, namely gcc-2.95 was used in one of the examples of Chapter 1 to illustrate the compilation process.

The reader should install CYGWIN 1.5 (or newer) for compiling and checking out the source code of examples in the case of PC running Windows. The source code can be compiled and checked on Personal Computers running LINUX Operating System. An HTML based documentation for the source code is also available as a part of the download.

Supplements for Instructors: PowerPoint slides, class-test quizzes with answers, chapter-wise references, and lab assignments.

Supplements for Students: Chapter-wise tutorials and self-test quizzes with answers.

The Yahoo Groups at <http://in.groups.yahoo.com/group/compilers2009> is a useful place to post any issues with regard to the compilation/execution of the source code, specific to your Operating System.

Acknowledgements

I am grateful to Prof Khodanpur, Retired Head of the computer science department at RV college of Engineering, Bangalore, for giving me the idea to write a book on compilers and reviewing my first output. I hope I have done justice to his expectations.

I am indebted to my family for all the sacrifices they had to make in order to fulfill my dream. Thank you Amma and Little fellow for everything. You have been a wonderful family. This book would not have been possible without your support, love and affection.

I would like to acknowledge the help of several people who have made a difference in my life. I owe a Big Thank you to Dr Aswath N Rao, N R Ramesh, M K Suresh Kumar, Priya Suresh, K Srinivas, R Sundararajan, J Swaminathan, Siby Abraham, S Krishna Kumar, Tinku Jose, Vijay Garapati, Sanjay Khodanpur, Shashikala, Srinath Rajaram, A Ramasamy, M Vijay Anand, V Balachandran, David Lobaccaro, Prakash Viswanathan and Srinivasan.

I would like to thank my first employers—Vinay Deshpande and Shashank Garg at NCORE Technologies, Bangalore—for giving me an entry into the software field. I want to thank my colleagues and friends at Wipro Technologies for all their help and cooperation.

I also want to thank Vibha Mahajan, Shalini Jha, Nilanjan Chakravarty, Surabhi Shukla, Surbhi Suman, Dipika Dey, Anjali Razdan and Baldev Raj at Tata McGraw Hill Education for their help and guidance in the course of making this book. A note of acknowledgement is due to the following reviewers for their valuable feedback.

Preeti Aggarwal

Punjab University of Technology (earlier Punjab Engineering College), Chandigarh

M Tripathi

Institute of Engineering & Technology, Kanpur

Shailendra Singh

Punjab University of Technology (earlier Punjab Engineering College), Chandigarh

Shalini Batra

Thapar Institute of Engineering & Technology, Patiala

K Umamaheshwari

PSG College of Technology, Coimbatore

R Prabhakar

Coimbatore Institute of Technology, Coimbatore

B N S Murthy*Nagarjuna College of Engineering & Technology, Bangalore***B I Khodanpur***R.V. College of Engineering, Bangalore***B K Sarkar***Birla Institute of Technology (BIT), Ranchi*

The readers of the book are encouraged to send their comments, queries and suggestions at the following email id—tmh.csefeedback@gmail.com (kindly mention the title and author name in the subject line).

V Raghavanraghavan.compilers2009@yahoo.co.in