

Roman Marynchak

SBCCL Compiler Internals

Ukraine, Ivano-Frankivsk

2010

Table of Contents

Preface	3
1 Python Architecture: the High-level Overview	4
1.1 What Python Does, and What It Does Not	4

Preface

A good preface is probably the most tricky thing to deal with. I remember when I was a student, I always spent a lot of time dealing with the article's preface. Following the obvious rule, let us start with the description of this book goals and scope.

The main goal of the book is to help us understand SBCL Compiler (later I will reference it as 'Python') architecture and design. Everybody has its own reasons to gain that understanding and knowledge – somebody wants to fix the compiler bugs, somebody wants to extend it with the latest algorithms and approaches from the modern compilers design field and somebody just wonders how this black box works. In any case, I hope that this book may be helpful for a reader whatever motives he or she has.

The book scope is quite large – at least it is planned to be so. SBCL is an industrial-level compiler, and there are a lot of things to look at. We are going to examine the compiler sources and see how it processes user's code from the REPL till the machine code generation. And we will not hurry. A good understanding requires a lot of time – and our vision of the compiler internals will evolve in time slowly. As a benefit there will be one beautiful day when you will feel that it is a time to do you own compiler fork with the-all-cool-things support, or at least you will be able to fix that annoying bug which was reported two years ago:)

This book is somehow specific: here one newbie is going to teach other newbies. Sounds like some nonsense... :) The thing is that I have a lot of experience with other compilers internals, and here I will try to apply my skills and do a reverse-engineering of Python. These reverse engineering results will be covered in the book.

Before the preface end, let us say few words about the style and text conventions. The style is connected with some interesting fact from psychology, which says that there should be at least 35% of freedom in the even most serious books. So I am going to write this book in the manner which is more close to a talk between friends than to an academic paper. This does not mean that we are not going to talk about complex stuff. We will talk about that, but I will try to explain it in the easiest way I could imagine. And I will insert some general, philosophical or funny sentences here and there – just to make this book more friendly and warm to the newbies than most academic books are.

Regarding the text conventions, they are simple: functions/macros names are given in **bold**, code snippets are images from the editor (to make the code look pleasant), figures are numerated and subscripted at the bottom.

Thanks to all members of SBCL community who encouraged me to start this writing. Personal 'thank you!' goes to reviewers: as for now I should name **Paul Khuong** here.

The last thing to mention: I am not a native English speaker, so I apologize for any grammar mistakes you may find in this book, and I will appreciate when you let me know about them.

Thank you for your time devoted to this book.

Roman Marynchak,
roman.marynchak@gmail.com

1 Python Architecture: the High-level Overview

1.1 What Python Does, and What It Does Not