

1、INTRODUCTION

1.1 Objectives

使用计算机语言是人和机器之间重要的一环。通过本文，我们希望读者更加意识到：

- 1，命令式程序语言，他们的语法和语义的特性，有问题的地方和歧义的地方，优缺点。
- 2，程序语言的翻译器，各种类型的翻译器（汇编器，编译器，解释器），翻译器的实现。
- 3，编译器生成器，帮助程序语言翻译器的自动构建的工具。

1.2 Systems programs and translators

现代计算机系统的使用者可以分为两种。一种是不去开发程序，而只是使用别人开发的软件。另外一种是会开发软件的人。第二种人通常会用高级语言来表达算法，用交互的编译器来编译软件。他们使用的一类工具称之为系统软件。

translator或者compilers接受用source language的文本，输出用另外一种语言表述的结果，通常是object 或者 target language.

一些新手在开发和使用高级语言编写的软件时，通常分不清编译和执行阶段。

一个translator(compiler)通常也是个软件，本身也用某种计算机语言编写，这种语言也叫host或者implementation language。现在很少有从及其语言开始编写的编译器。不过第一个translator不得不从机器语言开始写起，为一个新系统开发编译器也要了解机器语言和机器的架构。现在为新机器开发编译器，常常使用cross-compilation或者bootstrapping的技术。编译器通常不具备可移植性。

1.3 the relationship between high-level languages and translators

程序语言大体上分为高级的（比如pascal, fortran, C)和低级的，比如Assembler。高级语言分为imperative, functional, 或者logic(Prolog)

高级语言通常有以下优点：

- Readability
- Portability
- Structure and object orientation
- Generality
- Brevity
- Error checking

一种语言需要这几种要素：

- syntax --指定如何构成源代码
- static semantics -- 有哪些限制和范围
- dynamic semantics -- 有能力预测这种语言的语句执行的效果

compiler generator在现在会用来实现编译器。compiler generator用语言的正式的语法和语

义描述作为输入，将这种语言的编译器作为输出。