

编程语言 编程 计算机科学 编程入门 编程技术

如何自创一门计算机语言？

真心求教，目前正在看 操作系统,linux 方面的书

5 条评论 分享

搜索你感兴趣的内容...

首页 话题 发现

提问

注册知乎

登录

空明流转，计算机图形学、C++、编程话题优秀回答者·本人政治错...



148 148 人赞同

如果你仅仅有一般程序设计的基础，直接要设计一门语言是有难度的。结合我个人的经验，以下路径应该比较适合语言初学者：

1. 你需要有基本的编译原理常识。构造基本的编译原理常识，一方面来自于你对已有语言的使用经验，了解基本术语。比如我用C，那么我起码知道语言要素包括宏、表达式、语句、语句块、函数、指针等；我还知道C语言有编译、链接和执行三个阶段。这些基本概念对你宏观掌握你的学习进程是很有必要的。另外一方面编译原理的常识，要来自图书。比如龙书、SICP。在这一步，你得知道大部分语言的处理都要分为词法、语法、语义和代码生成四个阶段。每个阶段，分别是做什么的。
2. 了解具体的编译算法。了解到什么程度，取决于你使用第三方工具，还是需要自己从字符开始处理。个人建议，乔姆斯基文法体系、（扩展）巴克斯范式（EBNF），正则表达式，和LL(1)的递归下降分析法是必须要掌握的。对LL(k)，LR(k)要有概念。其中，四则运算表达式的分析是很好的练习。
3. 对语法的感受。初学者设计语言的难度有两点。**第一，不知道什么样的语法/语义是你需要的；第二，不知道你设计的文法能否实现。**如果你练习过递归下降法，应该已经有了基本的感受。这个时候可以试图实现你熟悉的语言。一些常用语言的Grammar Rule都是可以查到的。虽然这些语法，特别是C++，可能是上下文相关的所以用书上的办法难以实现，但是你可以实现它的一个子集。通过这样的练习，你能对你掌握的分析算法能完成什么样的工作就有了个大致的体验。
4. 深入了解你的领域，明确需求。你是需要一个类似于自然语言的脚本，还是只是一个表达式。经过3阶段的训练我认为你已经有了独立撰写语法的能力。可以写一个基础版本的出来。
5. 实现你的语言，特别是词法和语法部分。这一步最好能Log出尽可能多的信息，例如词列表并打印出分析树。
6. 当你有了分析树后，就可以去做语义分析了。对于初学者而言，语言的规模要尽可能小，这样语言的解释和语义分析可以合为一体，不用担心哪个部分算是代码生成，哪个部分算是语义分析。所以这一步最好你能把解释器也写出来。这一步的要求很简单，知道符号表是做什么的，学会归纳和推导类型，能正确的处理分支和循环语句（如果有的话）。
7. 设计一门类似于ASM的VM语言，可以是堆栈机模型，也可以是寄存器机模型。撰写代码生成从语法树上生成这个VM语言出来。并写一个VM的执行器。这一步看起来无从下手，实际上只要你肯写，不出一个礼拜你就知道怎么做了。一开始这个VM语言可以只用做整数的加减乘除就行了，以后可以慢慢加其他比较重要的特性，比如运行栈啦、代码地址啦，跳转啦等等。另外，我觉得把LLVM作为你的目标也是不错的选择，起码指令的解释不用自己费工夫了，LLVM的手册读起来也很容易。那个Tutorial写的特别好。

至此，你的语言就初步建立完成了。但是因为一开始你的能力有限，所以设计出来的语言是磕磕巴巴的。这个时候你可以选择：

- 向某个现有的语言靠拢；
- 如 @Belleve 所说，自己去了解现有语言的机制并进行权衡。这个时候什么动态/静态类型，函数是first class，类和对象才是你该考虑的问题。这个时候一定要多读、多用现有的语言。拜你的编译器经验所赐，你会站在一个不同角度上来对待现有的语言；
- 学习Functional Programming，这对你设计更好用的文法、更快的编写和修改编译器都有莫大的帮助。例如你可以使用Combinator在C++中以类似于DSL的方式配置语法；
- 剩下的就是在Language领域无止境的挖掘了。上下文相关的语法、语法糖的设计、运行时的辅助机制（例如async，await这些广义上的控制流语句），GC，类型推导和静态分析等等。@vczh在这点上更有发言权；

加入知乎

与世界分享你的知识、经验和见解

姓名

密码（不少于 6 位）

验证码



注册

已有帐号？ 登录

下载知乎 App

关注问题

710 人关注该问题

相关问题

换一换

面向对象编程的弊端是什么？ 192 个回答

怎样算是「风骚」的代码？ 96 个回答

为什么计算机能读懂 1 和 0？ 160 个回答

编程到底难在哪里？ 138 个回答

想成为计算机技术高手，一定要懂汇编吗？ 51 个回答

相关 Live 推荐



Python 工程师的入门和进阶



程序员的常见陋习



写作与编程



带你围观联合国摩洛哥气候大会



有趣的人体生物钟

- 编辑于 2014-03-06 6 条评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

一门语言在程序语言的角度上是不是好，跟他的library毫无关系，跟他能够写出怎样的library才有关系。一个library还不存在，有可能代表他写不出来，也有可能代表只是没人去写。当你试图创造一门语言的时候，你就不能以大众的标准来衡量他，比如说【我大python有numpy！】。其实哪个语言不能有numpy？只是没做罢了。所以在numpy这一点上，大部分语言都是一样——能做。你关心的就是这

些，能，还是不能。而不是有，还是没有。譬如说【我大go有channel!】——这是微软很多年前的后来改名叫midori的C#操作系统项目早就给C#实验性添加的东西就不说了，F#的computation expression也好，erlang的actor也好，哪个不比go的channel强？再说了，Java的库为什么用起来就是不如C#的舒服，这完全是因为Java的语法导致的，而不是人类的想象力导致的。

好了，当你能够准确的指出上面所说的所有语言的任何功能的好处、坏处、解决的问题、带来的坑之后，你大概就可以站在巨人的肩膀上来设计一门语言了。然后你要关心的问题就变成了：

- 1、我想用这门语言来解决什么问题（你说他是general purpose的也可以），因此需要什么样的语法，才能提供称手的library。
- 2、怎样实现他才够牛逼。
- 3、到底有没有比别的语言好的地方（否则就跟go一样成为反面教材了，新的都不是好的，好的都不是新的）。
- 4、学习设计指令集和虚拟机，或者JIT（也就是说要编译成x86）。

当你学会了设计，也学会了实现的时候。估计也差不多熬成大牛了。

编辑于 2014-03-06 31 条评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲

温悦, I am pf_miles@alipay.com

1

1 人赞同

▼

很容易，每个月、每年都有无数种计算机语言被创造，光是出自计算机系本科生编译原理大作业的就不少；
所以，就题论题的话“如何自创一门计算机语言”其实你只需要按照教科书上的做就行了；
不过若说“如何创造一门优秀的编程语言”或者说“流行的编程语言”那是另一些广阔的话题

发布于 2012-01-13 1 条评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲

冒泡, IT

0

▼

自创语言难点不在于实现技术，这些慢慢学总有的，在于设计，你的设计对需求方（你自己，或者其他）是否好用
[xtlisk的专栏](#)
这里有我的一个系列blog可以参考下

发布于 2016-03-08 添加评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲

知乎用户, 计算机科学算是一点业余兴趣爱好

0

▼

选一个编译器编译器来学吧（非错字
比如yacc或者yacc或者yacc.....

发布于 2012-06-12 1 条评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲

Kirisame Maris, 磕了它好久，都没磕出血来。O,O

14

14 人赞同

▼

[code.google.com/p/grass...](#)

给你个乐子，亦可借鉴。

发布于 2011-09-26 5 条评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲

李凯东, 9层爬到5层的经历者

2

2 人赞同

▼

开始之前是要做语言设计，是脚本语言、标记语言、命令式语言亦或者是全功能语言。
写 Backus-Naur Form 范式（需要编译原理和离散数学）

接下来
关键词 分词（还是状态机、形式逻辑等（这个级别就不是用正则表达式了））
语法分析产生语法树（状态机理论、数据结构、深入了解递归原理）
根据语法树变成中间语言（继续编译原理）
然后在把中间语言针对某种cpu编程（汇编）

第 4 页 (共 6 页)

▲ **TsingsonQin**, 亲子启智/越野/口琴/大画幅
0
个人认为，自创一门计算机语言，有三种方法：

▼

1. 自创一种计算机，这样能自由的开创一种全新的计算机语言----- 重定义“计算机”！！
2. 针对某个领域或某一类问题，全新开创一种语言，仅为解决这一类问题。（这时候，你可以少考虑什么编译原理、或者参考别的语言）
3. 当然了，上面说的所有答案都比较符合你的“想法”----- 走前人走过的路，加入你的想法。

发布于 2015-03-21 添加评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲ 知乎用户, 哲学 机器智能 C++
1
1 人赞同

▼

看了一下答案，这么多人都从技术上去分析，怎么没有人提到需求？单从技术上看，设计一门计算机语言其实有蛮成熟的技术路径，只是时间和工作量的问题，但你要从已经存在的众多语言中找出需求差异，找到你新语言的存在价值，却是一件很难很难的事

发布于 2014-03-06 添加评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲ **NosicLin**, 编程要的是速度与质量
1
1 人赞同

▼

我的毕业设计就是开发一门面向对象的脚本语言，最近正在加班加点的写找码，我把所有的文档都放到了博客上面了，链接为blog.csdn.net/forwardli...

希望有喜欢编译原理一起来参与开发

编辑于 2012-03-01 1 条评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲ **讨厌复杂**, 全能高级程序员
0
很简单，从C开始。

▼

编辑于 10:01 添加评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲ 知乎用户, 计算机语言解释器开发
0
设计语言, 通常有很明确的目的, 要门现有语言无法满足需要, 要么能够在某一方面超越同类软件. 如果

▼

只是为了设计而设计, 验证老师教的东西, 我认为没有任何意义. 编译原理和大学里教授的通用语言设计知识, 已经过时了, 新的语言设计架构和思路很多. Perl6 的语言架构就是最新的.

发布于 2016-08-26 添加评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

▲ **PENG QIN**, 学生
0
最近老师就在让我们自己做一个语言。。。。

▼

鉴于我是学渣，我只能告诉你一些我知道的信息

compilation 是很重要的 lex和yacc要熟练掌握 我们用的是ocaml 这种函数式的编程挺有意思

Pierce—book是我们学习的主要内容

祝你好运 我也在尽量理解

发布于 2016-03-18 添加评论 感谢 分享 收藏 · 没有帮助 · 举报 · 作者保留权利

更多

我来回答这个问题

写回答...

我要回答