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

加入知乎

5世界分享你的知识、经验和见解

姓名

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 应该比较话合语言初学者:
  - 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. 对语法的感觉。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觉。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领域无止尽的挖掘了。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;



下载知乎 App

关注问题

710 人关注该问题

#### 相关问题

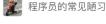
换一换

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



写作与编程



有趣的人体生物钟

• 最后也是最重要的,**充分理解你的需求**,才能在诸多语言要素间做出取舍。

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

▲ Belleve , 编程、JavaScript话题优秀回答者・I focus on Type. ...



- 14 14 人赞同
- 第一件事情是先把语言设计出来。设计语言(尤其语义)需要的知识是形式语言理论、类型论等,而非编译原理。在这阶段要考虑的东西包括:
  - 1. 编译还是动态解释? (JIT 也算编译哦)
  - 2. 静态类型还是动态类型?
  - 3. 函数是不是 first class?
  - 4. 是否要引入宏?
  - 5. 是否要引入对象?如果是,是否要引入类?它们之间关系如何?
  - 6. 等等等等......

等你设计完语义就要设计语法。语法设计很繁琐,因为要处理歧义。 等上面的都做完,就可以写编译器了。

编辑于 2014-03-06 2条评论 感谢 分享 收藏·没有帮助·举报·作者保留权利

vczh, 《C++Primer 5th》强势审校



# 224 人赞同

你要学牛顿站在巨人的肩膀上看问题(不过不是让你去学他怎么骂胡克)。因此在设计一门语言之前,你应该把目前重要(不一定代表流行)的语言全部学会,包含但不限于

C#: 面向对象做得最好的,而且很多语言的yield return和await就是从这里抄的,而且人家还有披着ling皮的monad。

Haskell: 包含了最多纯函数式和类型推导的内容

erlang: 什么是actor

prolog: 原来写程序还可以通过描述一个对象(不是以方程的形式)来让计算机自己求解

lisp: 你得从这里学会continuation passing style

delphi: 跟C#差不多,但是人家是用引用计数的,而且不对class做任何垃圾收集

ruby: 人家的DSL做得多好

perl和VB.net : 主要是学习如何把regex/xml什么的完美的嵌入到语法里面 ↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓

C++: 著名的多范式语言(如果你连小小的模板元编程都搞不定我觉得你还是不要试图创造语言了,没

戏的), @吴野 菊苣最近还在持续更新C++模板的教程 github.com/wuye9036/Cpp...

F#:告诉你函数式也是可以不纯并且面向对象的,而且作为一门函数式语言,完全贯彻了【我自己也能做控制流】的思想,完爆上面的yield return和await——这个功能叫computation expression——如果你牛逼甚至可以写library来完爆erlang

你还可以来看一下我做的 github.com/vczh/tinymoe 也是一门自带cps变换、语法上的描述能力跟 lisp的宏毫无区别的、严肃的编程语言,只是长得跟别人略有不同。哦,对了,我还实现了multiple dispatch,好像我还没有在什么靠谱的语言里面看到如此重要的feature,能与之一比的也就只有ruby的open class了,至少它可以用迂回猥琐的方法来实现等价于multiple dispatch的东西。

当然,站在程序设计语言的角度上,有一些大家很喜欢的语言,纯粹是当做反面教材来用的。反面教材的标准就是,所有的feature都没有一个是站在所有语言的顶端的(看我上面的例子明白什么是"至少xxx功能比别的语言都好")。包含但不限于:

python, javascript, go, C

\_\_\_\_\_

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,譬如说【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
- ▼ 他人)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) 这只是基础的,上层还需要做对外sdk, 那就更多了

基础数据类型

字符串处理

io处理

内存管理

网络通讯处理

还可能一些组件库

事件

ui绘制库

等等 等等

还有外围的

打包、安装、资源管理等等

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

- 知乎用户, 软件工程师, 邮箱从业者
- 1 1人赞同

0

推荐一本书《自制编程语言》

amazon.cn/%E5%9B%BE%E7%...

发布于 2014-03-06 添加评论 感谢 分享 收藏 • 没有帮助 • 举报 • 申请转载

AlephAlpha, (\*\_\*)



▼ 什么的。还可以增加一些你喜欢的功能或者奇怪的约定。如果有兴趣的话,还可以把 brainfuck 语言推 广到高维……

很多 esoteric programming languages 都是这么来的。

发布于 2013-10-16 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, 码农
- 0 如果你想了解编程语言是怎么工作的,可以按照@空明流转 给出的知识点去学。但是你要自创一门编
- ▼ 程语言,首先应该问自己这门编程语言能够解决什么问题,相比其他编程语言有什么优势?

在自创之前应该先去了解主流编程语言的特性和优势,了解编程语言的发展历史,不要闭门造车。

发布于 2014-04-26 添加评论 感谢 分享 收藏・没有帮助・举报・作者保留权利

- ▲ Cosmia Fu,会写Python的Haskell原教旨主义者
- 0 像牛顿一样站在巨人的肩膀上,首先你得先学会骂微软,啊哈哈哈哈 @vczh

要黑下微软才能显的逼格高,我是认真的

发布于 2014-04-14 添加评论 感谢 分享 收藏・没有帮助・举报・作者保留权利

- ▲ Irons Du, 跨平台C++(Lua)网络库 ...
- 1 1人赞同
- vczh和空明都答得非常之好。(排名不分先后)。

我的回答就是\*现阶段\*可下载[[游戏脚本高级编程]源码,调试它。(其中基本包含了空明聚聚提到的内容)

编辑于 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

- ▲ **孙立**, 不惑好奇
- 0 建议先看看SICP。
- ▼ 然后再看看DSL。

编译原理可能一下跳入细节,事倍功半。

发布于 2013-10-20 添加评论 感谢 分享 收藏・没有帮助・举报・作者保留权利





	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:			
•	17,000, 100 11014100411, 62-1177/4			
	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			
1	知乎用户, 哲学 机器智能 C++ 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			
1	NosicLin,编程要的是速度与质量 1 人赞同			
•	我的毕业计设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			
0	<b>讨厌复杂</b> ,全能高级程序员 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	知乎用户,计算机语言解释器开发设计语言,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.			
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			
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论知识, 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.			
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			
•	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			
•	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, 学生最近老师就在让我们自己做一个语言。。。			
•	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			
•	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			
•	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			
•	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			

找来回	答这个		题
-----	-----	--	---

写回答...

我要回答

© 2016 知乎

刘看山 · 移动应用 · 加入知乎 · 知乎协议 · 联系我们