

编程语言

如何学习 X

编程

C (编程语言)

关注者

10,065

被浏览

2,615,298

如何学习 C 语言?

关注问题

写回答


邀请回答

好问题 76

12 条评论

分享

...

 登录后你可以

不限量看优质回答

私信答主深度交流

精彩内容一键收藏

登录

查看全部 154 个回答

 黑马程序员

已认证的官方帐号

5,312 人赞同了该回答

有一句说的特别对“C生万物，编程之本”！C语言在计算机体系占据着不可替代的位置！

C/C++语言在计算机体系占据着不可替代的位置



如果你不知道如何学习C语言，那么，跟着下面的学习路线图来学吧，附带学习视频哦！





继续浏览内容

知乎

知乎

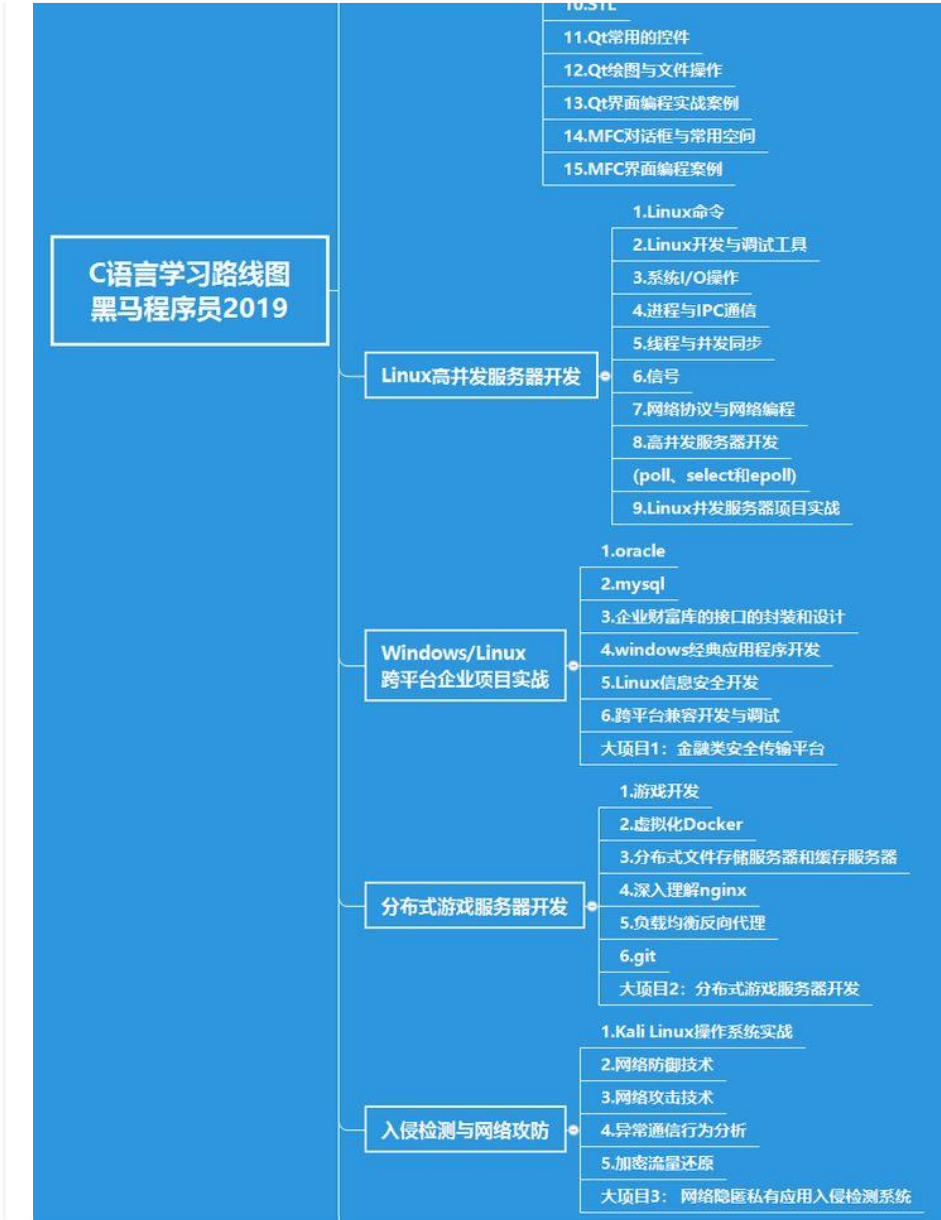
发现更大的世界

打开

Chrome

Chrome

继续



C语言配套学习视频:

1小时透彻理解C语言指针

完整视频:

指针如此简单--1小时透彻理解C语言指针

yun.itheima.com/open/351.html2005stt

配套资料: pan.baidu.com/s/14bnf2W... 提取码: 2vk4

继续浏览内容

知乎

知乎

发现更大的世界

打开

Chrome

继续

配套资料: pan.baidu.com/s/1jhJpIS... 密码: ov17

C语言数据结构 培养架构师思维

完整视频:

C语言数据结构 培养架构师思维

yun.itheima.com/course/428.html?2005stt

配套资料: pan.baidu.com/s/1C4cm3t... 密码: 4wi6

最适合自学的C++基础视频_深入浅出出版全套资料

完整视频:

C++教程_C++教程视频_C++教程下载

yun.itheima.com/course/10.html?2005stt

配套资料: pan.baidu.com/s/1brlk0A... 密码: n66u

匠心之作 C++从0到1入门学编程

完整视频:

C++| 匠心之作 从0到1入门学编程

yun.itheima.com/course/520.html?2005stt

配套资料: 链接: pan.baidu.com/s/1PbkG54... 提取码: q0i4



完整视频:

使用策略模式实现游戏技能系统

yun.itheima.com/open/232.html?2005stt

配套资料: pan.baidu.com/s/1hrtys5... 提取码: mpxa

linux典藏系列之 网络编程

完整视频:

linux典藏系列之 网络编程

yun.itheima.com/course/429.html?2005stt

继续浏览内容

知乎

知乎

发现更大的世界

打开

Chrome

继续

C++核心编程_打造你的核心编程技能

yun.itheima.com/course/427.html?2005stt

配套资料: pan.baidu.com/s/1F_tvZX... 密码: cka8

C++关系型数据库编程 | 4天从入门到精通

完整视频:

C++关系型数据库编程

yun.itheima.com/course/435.html?2005stt

配套资料: pan.baidu.com/s/1YETuEI... 提取码: i1qk

C++ STL 高级教程 | 深入浅出版

完整视频:

C++ STL 高级教程 | 深入浅出版

yun.itheima.com/course/434.html?2005stt

配套资料: pan.baidu.com/s/1sli_-2... 密码: cj6j

步步打造-frameQT自定义界面

完整视频:

步步打造-frameQT自定义界面

yun.itheima.com/course/433.html?2005stt



QT从入门到实战完整版

完整视频:

QT从入门到实战完整版

<yun.itheima.com/course/539.html?2005stt>

配套资料: <pan.baidu.com/s/1o2MzfY...> 提取码: xapf

MongoDB从配置到编程

完整视频:

继续浏览内容

知乎

知乎

发现更大的世界

打开

Chrome

继续

完整视频:

零基础学会Linux系统编程

<yun.itheima.com/course/430.html?2005stt>

配套资料: pan.baidu.com/s/17E_5UI... 密码: 3l77

半小时快速搭建网络攻防环境-Kali操作系统

完整视频:

半小时快速搭建网络攻防环境-Kali操作系统

<yun.itheima.com/course/436.html?2005stt>

配套资料: <pan.baidu.com/s/1yLqeSZ...> 密码: 49ds

C++与分布式数据库开发

C++与分布式数据库开发

<yun.itheima.com/open/170.html?stt>

资料: <pan.baidu.com/s/1o7SwLr...> 密码: 2g5t

c++与安全协议开发

完整视频:

C++与安全协议开发

<yun.itheima.com/open/191.html?2005zpzp...>



C生万物之游戏内存劫持

完整视频:

C生万物之游戏内存劫持

yun.itheima.com/open/524.html?2004stt



配套资料: pan.baidu.com/s/1BYDSXZ... 提取码: 9bzz

金山游侠开发

完整视频:

继续浏览内容

 **知乎**
发现更大的世界

打开


 **Chrome**

继续

完整视频:

C++干大事系列之游戏篇: Qt飞机大战

yun.itheima.com/open/478.html?2005stt




配套资料: pan.baidu.com/s/11XD9Ma... 提取码: 9zuz

QT跨平台小游戏开发—黑白棋

完整视频:

QT跨平台小游戏开发-黑白棋

yun.itheima.com/open/101.html?2005stt




配套资料: pan.baidu.com/s/1oA4skN... 提取码: e2wa

【毕设项目】使用QT实现跨平台开发“快播”播放器

完整视频:

【毕设项目】02使用QT实现跨平台开发“快播”播放器

yun.itheima.com/open/23.html?stt




配套资料: pan.baidu.com/s/1dYy_wk... 密码: wylm

更多C++学习教程

C++免费教程_C++免费教程视频_C++教程免费下载_最新

yun.itheima.com/open/c-6.html?stt





yun.itheima.com/course/55-0-0-0.html?stt

编辑于 2020-05-25

赞同 5312 139 条评论 分享 收藏 喜欢 收起

更多回答



Meta

647 人赞同了该回答

这个应该是几个月前的提问了，但是C语言的问题过十年问也不晚。

一、C语言是最值得学习的语言

继续浏览内容



打开



继续

... 恰当的语言，一切都在掌握之中，一切荣耀归属于coder，一切错误也归罪于coder。

3. 学习C语言的同时，就自然而然的理解了硬件与操作系统的浅表一层。比如int有几个字节的问题，已经是QQ加群验证是否是程序员的一个标准了（笑），这从侧面说明了学习C语言时确实会关心底层软硬件的实现。C语言简单的基于值类型的数据类型体系（引用靠指针，指针本身也是值类型），保持了和硬件底层的一致性（即内存、缓存、寄存器只能保存值），不会出现其他语言比如C#那样既有值又有引用反而对理解底层实现制造了障碍。

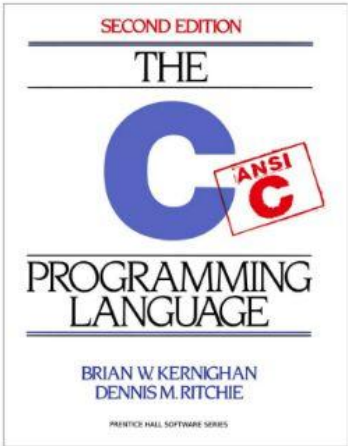
4. 从某些角度上讲，C语言从设计上来说并不是为了让我们更好的写逻辑代码而设计的，它真的是“只能是这样”，没有对错好坏之分。因为计算机体系结构就是这样的、汇编就是这样的，所以C语言大致也只能是这样的，谁也不能随意捏造它。它足够简单，评论它的设计优劣意义不大。

废了这么多口舌，是为了坚定思想，端正态度，只有抱着“最好”的态度，才能排除杂念，勇往直前。时代在进步，某些东西易逝，而另一些东西持久。很多编程大师都用亲身经历告诉你，C语言永远值得用最纯粹的态度去学习。

二、学习C语言的初步路线

我们说学习任何东西，都有先把书读厚，再把书读薄的过程。也就是说，先学习大量知识和细节，然后总结归纳、提纲挈领，就彻底掌握了这门知识。

而学习C语言，一开始就不那么厚：)。大道至简，只需一本：



吧)。不用买一堆入门到精通、高级C、本质论、Primer互相参照。C语言的教材、参考书都是这薄薄一本。

C语言的语言本身很容易“精通”，一旦你理解了指针、函数指针等问题，基本知识方面就横行无阻了，很少会遇到完全没见过的语法。

C语言基本学习过程真的没有太多好讲的，假设每天学习两小时，每周学习5天，只要动手多写例子，把例子保存好随时翻看，那么C语言的语法基础很快就能全部过完一遍，估计半个学期的时间就能看完。难点可能是在指针的使用，这需要多看资料、多画图、多写代码尝试，给自己点时间强行突破即可。

但是问题也在这里，由于宏和指针的存在，万事皆有可能。很多经典的C代码，真的是没见过之前根本不知道还能这么用，举个入门级的例子，Linux内核中的一个通用链表：

```
// 定义链表节点结构
```

继续浏览内容

 **知乎**
发现更大的世界

打开

 **Chrome**

继续

```
// 创建链表

#define LIST_HEAD_INIT(name) { &(name), &(name) }

#define LIST_HEAD(name) struct list_head name = LIST_HEAD_INIT(name)

// 判断链表是否为空

static inline int list_empty(const struct list_head *head){return head->next == head;}

// 运行时初始化链表

#define INIT_LIST_HEAD(ptr) do { (ptr)->next = (ptr); (ptr)->prev = (ptr); } while (0)

// 我自己写的测试函数，方便读者阅读：

int main()
{
    LIST_HEAD(my_list_head);

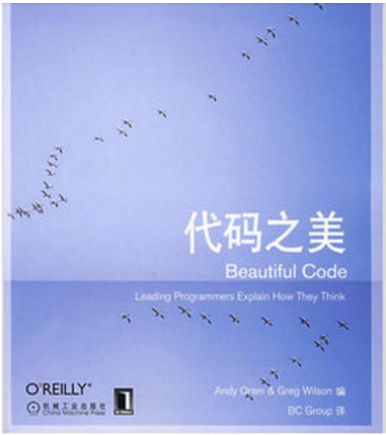
    printf("列表是否为空 %d\n", list_empty(&my_list_head));

    return 0;
}
```

C语言写的经典代码经常会用这种“过度简洁”的语法，初学之后会看得云里雾里的。如果你能看懂，又会有一个更大的问题——这个list_head结构体里面，没有数据。自定义数据应该放在哪呢？这又需要深一步研究了。

稍微高深点的比如任意动态语言的源码，比如Lua的源码，对于刚学懂了C语言的同学来说，真是每个字都懂，连在一起就不认识的感觉。还有一些经典算法，比如Subversion的源码，都很值得深入分析，只要理解一个模块，就会收获巨大。具体的例子这本书里提到好几个：





赏析一下书中前几个问题：

继续浏览内容

知乎

发现更大的世界

打开

Chrome

继续

可能是从这本书开始，本书封面设计让我想起有个哥们说，对小白子们来说，那本书非常值得一读，可以迅速意识到自己的不足并迅速成长。本书建议先找电子版试读。

PS：关于C语言开发环境

- 1、Windows平台，请使用Dev-C++。不要用Visual Studio，对C语言支持不完整。
- 2、安装Linux虚拟机，桌面版本的，比如Ubuntu，现在貌似“深度Linux”挺火，没试过。

三、关于进阶

C语言的进阶会遇到两个主要问题：

- 1、明白基础以后，缺少大量练习。而大量练习本身会非常费时间，因为在顾及语法的同时要顾及逻辑问题，捉襟见肘。这时候的初学者，逻辑抽象能力和语法使用都是半吊子，非常难受。
- 2、阅读代码方面，C语言的成熟工程代码一般会比较冗长，函数中间夹杂着很多宏和指针操作，我阅读的感觉是读后面忘前面，非常难入手。如果不明白某个函数的功能，强行将其读懂非常费时费力，容易放弃。

这样一来，学完基本知识之后的进阶之路真的非常陡峭。

我个人大胆给出C语言进阶的一个相对缓和的路线：

- 1、实现链表、可变量数组、哈希表等基本数据结构，实现方便的string结构体。然后大胆的用在自己写的程序中。不要怕写的有BUG，也不要怕效率不高。现代编程中容器是非常关键的武器，没有容器寸步难行。而编写容器时会练到封装、指针、内存分配，都是C语言进阶时极其重要的方面。
- 2、如果工作学习中用到C语言就再好不过，比如现在很多本科、硕士毕业论文都要用C语言实现，那么这时候抓住机会多练练，是最好的。
- 3、尝试ACM竞赛题，ACM竞赛和OnLineJudge平台，比如

tyvj.cn/



续下去，哈哈。不过除非你兴趣就是算法本身，那么刷30道题基本就可以了，毕竟这种刷题会猛烈地提高算法能力，但是对架构能力的提升几乎没有任何作用，对大多数人来说不建议刷太多。

4、非常重要的私货：如果除了做练习题以外没有练习机会，那么请把C语言放一下，快速学习Python（首选）、Lua（次选）、JavaScript（或TypeScript更好）等能够立即实现功能效果的语言。学习好的动态语言有两方面好处：

第一：表层来说，能迅速提高逻辑代码的编写能力。用C语言构思半天才能写一点的功能，用动态语言会快速的多，动态语言将我们从内存分配、容器等基本问题中解脱出来，直接实现我们想要的效果。用动态语言实现之后，可以用C语言仿写这些逻辑代码，相当于按图索骥，直达目的。

第二：深层来说，较高级的语言对底层的语言编写有强烈的指导意义。我个人在工作中使用C++用了很长时间，但是总感觉遇到瓶颈无法突破。直到我学习了Python，写了一两个比较复杂的工具，从另一个角度突破了逻辑设计的屏障，才感觉到编程水平有了明显提高。

第三：在掌握了Lua或Python后，再找资料一边学习一边看Lua和Python的C源码，对提高C语言水

继续浏览内容

 **知乎**
发现更大的世界

打开

 **Chrome**

继续

四、总结

写了很多，感觉再写下去个人私货就太多了，并不客观。

总之C语言的学习是知识点少、难点多。上面针对初学提供了一些建议，进阶方面写了很多个人的心得体会，希望能给学习C语言的初学者和进阶者提供一些帮助。

如何学习C语言这是个挺重要的大问题，读者若有意见和建议，可以讨论交流，有问题我可以修正。

发布于 2017-09-20

▲ 赞同 647 ▼

● 34 条评论

➦ 分享

★ 收藏

♥ 喜欢

收起 ^



编程指北  

腾讯科技（深圳）有限公司 后台开发工程师

372 人赞同了该回答

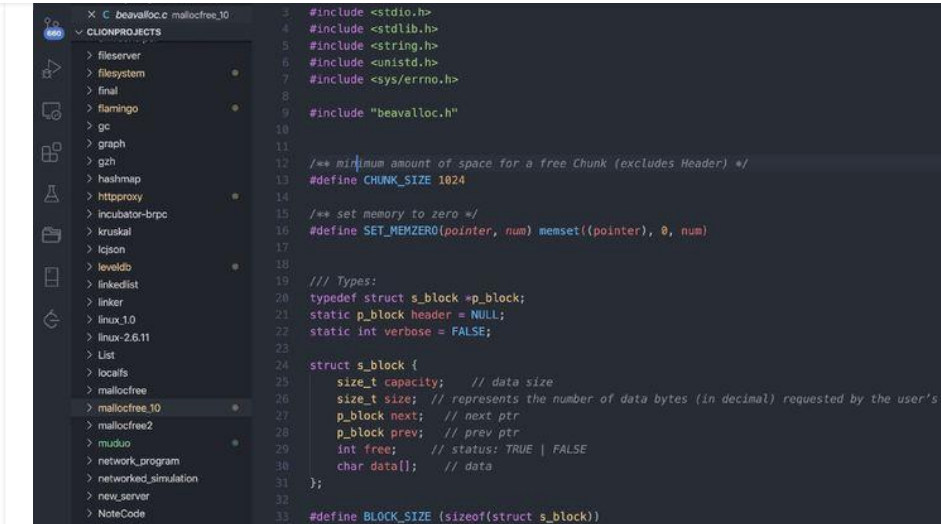
作为一名写了好几年 C 语言，也给很多留学生讲过 C/C++ 的学长来回答下吧。

我用 C 语言写了很多小工具，比如malloc、free，挂载到fuse上的filesystem，简易版svc等等。

总的来说，有上万行以上吧，对于指针、内存这些C语言关键点，也还算掌握得不错。

跟着我的方法学习，C语言一定没问题~~





继续浏览内容



打开



继续

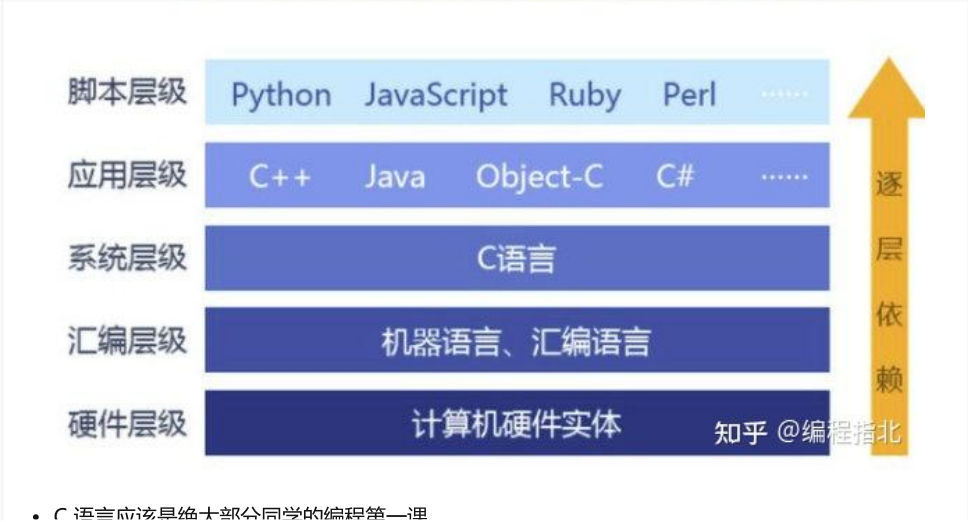


获取方式可以看看这篇文章，附带了电子版的PDF下载链接，赶紧收藏起来吧：

[C/C++必读书籍推荐| 附下载方式](#)

再老生常谈下，强调下 C 语言的重要性哈：





继续浏览内容



打开



继续

其实，这是很多初学者都会踩到的一个坑！C语言本身是一门很简单的语言，提供的实用功能不多，大部分要借助操作系统、第三方库、单片机来完成。也就是说，只学C语言基本什么也做不了，也基本找不到工作。

C语言是一门通用性的语言，并没有针对某个领域进行优化，在实际项目中，C语言主要用于较底层的开发，例如：

- Windows、Linux、Unix 等操作系统的内核90%以上都使用C语言开发；
- 开发系统组件或服务，用于支撑上层应用；

既然C语言的应用这么多，为什么很多读者觉得它什么也做不了呢？

我们先说一个概念，就是库（Library）。库就是编程专家写好的代码，我们可以拿来直接使用，这样能够节省开发成本，提高开发效率，并且库代码的执行效率、严谨性、安全性和规范性要明显优于我们自己编写的代码，市场上有很多优秀的库，有的收费，有的免费，我们要善于利用这些库，尽量不要重复造轮子。

库一般分为两种：

- 编程语言的开发者在开发编程语言的时候，一般都要预先写好常用的代码，或者说常用的功能，例如输入输出、数学计算、文件操作、网络操作、日期时间、错误处理、字符串处理等，这些由官方编写的库称为标准库（Standard Library），它们随编程语言一起发布，可以认识是编程语言的一部分。
- 有一些组织机构或者个人也会开发一些库，有的是为了盈利，有的是业余爱好，有的是本公司正在使用的代码，开源出来造福人类，这些库称为第三方库（Third-party Library）。

如何学习？

一、一图胜千言

如果你不想看冗余的文字，直接看我画的这个思维导图即可：





继续浏览内容

知乎

发现更大的世界

打开

Chrome

继续

C 语言本身只包含了编程语言最基本的语法，比如变量、if、else、for、while、枚举、结构体等，外加一个指针。

但是为什么大家都觉得难呢？

说下我大一时觉得难的原因吧，主要两点：

1. 库太少，太原始

用 C 语言写个稍微复杂点的项目，你需要上来先把数据结构、常见算法先实现一遍，更别说字符串，序列化这些了。

比如 Redis 里 SDS、LIST、ZSET、HASH等。

这其实对新手极其的不友好。

想做点有趣的东西，来，先写个链表？

而在 Java 里直接 New 一个 LinkedLists、HashMap，简直不要太高效。

再到后来，学了 Python，才发现这才是编程的高级形态，写个爬虫直接 import xx，几句话就完事了。

而在 C 语言里，你需要裸写 Socket 发网络包，还要解析 HTTP 协议，还要序列化 JSON，处理 HTTPS 这些（当然，你也可以导入库，但是由于 C/C++ 弱鸡等同于没有的包管理，也挺麻烦的。

这就是新手学 C 容易被劝退的原因，做不出好玩的东西，全是一堆黑框框。

2. 内存、指针

另外，常另 C 语言初学者感到头疼的就是指针了。

指针其实本身不难，就是变量的地址嘛。

但是问题在于地址是个什么东西？





继续浏览内容

 **知乎**
发现更大的世界

打开

 **Chrome**

继续

当然是需要地址啊，地址就跟快递柜上的编号一样，而快递柜格子里就相对于内存中真正存放的内容。

记住这张快递柜的照片，其实指针你就理解了。

什么是二级指针？

就是快递柜格子里放的是另外一个格子的编号，对应到 C 语言举个例子：

```
int a = 10;
int *pa = &a;
int **ppa = &pa;
```

ppa 所指向的内存存放的内容是 pa 的地址，pa 存放的是 a 的地址(就不画图了，脑补)

那你要问了，那指针和二级指针以及更多级的指针有什么区别呢？

为什么二级指针就要两个 ** ？

很简单啊，不用两个 ** 的话，你怎么告诉编译器，这个地方放的其实是另外一个内存的地址？

这样编译器才能去做语法检查，不然谁知道你这是放的变量地址还是另外一个指针的地址啊。

更加详细的内容其实我在这篇指针的文章中讲过：

为什么指针被誉为 C 语言灵魂？

不过现在站在我的角度，觉得指针很简单，似乎很好理解，也许是知识诅咒的原因。

知识诅咒：指的就是一旦我们自己知道某样东西，我们就会发现很难想象不知道它的时候会是什么样子。

后来接触到 Java，如同发现了新大陆一般。

在 C 语言里，申请一个内存，你得随时记得在合适的地方释放。

释放得不合适？对不起 core dump 随时等着你。



乱写指针? 对不起, 内存越界等着你~

栈内存越界? VS下, 烫烫烫等着你~

这些东西想必是每一位 C Programmer 的家常便饭了。

而在 Java 这类更加高级语言中, 直接 new 一个就行了。


好了, 扯淡完毕, 在这里给出一条系统化学习 C 语言的路线吧:

第一阶段: 初学少看书, 多看视频

这真的是血泪教训, 我大一会傻乎乎的去把课本看一遍, 然后做课本后面编程练习题。

学得那叫一个艰难呀, 大家都知道, 课本为了保留其严谨性和全面性, 往往都具有一个特点, 那就是「不说人话」。

继续浏览内容

 知乎

发现更大的世界

打开

 Chrome

继续

```
hello.c:3:5: error: implicitly declaring library function 'printf' with type
      'int (const char *, ...)' [-Werror,-Wimplicit-function-declaration]
      printf("hello, world");
      ^
hello.c:3:5: note: include the header <stdio.h> or explicitly provide a declaration
      for 'printf'
1 error generated.
<built-in>: recipe for target 'hello' failed
make: *** [hello] Error 1
```

知乎 @编程指北

比如这种很初级的报错, 但是初学者往往看到就慌了。

遇到这种情况, 我的建议是仔细阅读报错的提示, 解决方法往往就在这些 note 中, 如果实在解决不了, 就复制 note 去 Google 上搜。

这个阶段你需要三份资料:

- 一是补 CS 基础概念, 也就是计算机导论
- 二是一本 C 语言的好书。
- 三是一个优质的 C 语言视频

对于每一点, 我都只推荐一份我觉得最合适的, 避免你陷入选择困难症:

- 计算机导论

CS 50, 这是哈佛的一门计算机入门神课, 忘了是在大一下还是大二看的了, 当时觉得如获至宝。链接如下:

<https://cs50.harvard.edu/college/2021/spring>

让我现在还记忆尤新的一个点, 是这个老师讲到 binary search 的时候, 直接举了查字典这个过程, 然后现场手撕字典。。。





另外，这门课是用 C 语言作为教学语言的，非常难得。

因为现在国外很多 CS 名校的导论课都会采用类似 Python、Schema 这样的语言。

继续浏览内容

知乎

发现更大的世界

打开

Chrome

继续

https://www.icourse163.org/course/ZJU-199001

说到这，我不得不多说两句翁凯老师了，我看了好几门他的课，最直观的感受是：

讲解透彻、声音好听，而且特别注重细节，贴一张浙大匿名教室系统关于翁凯老师的评价吧：

143个评论

人气评论 最新评论

上翁凯老师的课收获很大，老师上课也很有趣，很能讲故事。即便是上一门基础编程语言课，所学到的也不仅仅是一门编程语言，而是对计算机科学技术更综合的认识。

发布于 2017.09.02 举报

120

翁凯老师！如果从最初的编程语言基础课都是这样的老师的话一定能学到很多东西。

发布于 2015.03.14 举报

52

不会照本宣科，不会念ppt，即便ppt做的很简洁有趣，不会空谈理论.....是一门可以水但你不会去水的课

发布于 2015.03.15 举报

48

听MOOC最大的感想就是.....他的两只剪刀手Y(^_^)Y

发布于 2017.03.20 举报

41

我在网易听的 也现场听过老师的课 非常好 老师真的是想把计算机内部的原理将给大家 让大家更透彻的理解一些语言 还有 老师的声音简直让人听了停不下来！！

发布于 2015.03.15 举报

39

脑残粉怒赞！！！！他哥哥的课一个都不放过！！

发布于 2015.03.19 举报

32

第一次上课就爱上这个老师了，上课一点也不枯燥，瞬间爱上java课。变身脑残粉。

发布于 2015.03.23 举报

23

翁老师虽然教计算机，但是知识面非常广博，而且声音超好听上课非常有乐趣！

一句话，翁凯老师是真的想把计算机内部原理、编程语言以深入浅出的方式教给大家。

- C语言书

《C程序设计语言》，一本被称作 C 语言圣经的书。

复杂的声明。

不过要看懂里面的每一个例子，也是有一定难度的，需要有一点点基础，但是不妨碍作为入门书，可以多看几遍。

对了，一直说学编程要多写，但是很多初学者也不知道写什么。

在这里，我给出一些学完 C 语言基本语法后，可以练手的例子：

- 链表

就用 C 语言把链表翻来覆去的写，从最基本的链表插入、删除、单向、双向、环装链表。

到链表反转、合并、分割等。

别看基础，但是很多同学到大四了，可能都写不对，这里考察是否足够细心、逻辑是否缜密。

继续浏览内容



打开



继续

第二阶段：搞懂内存，看书、写代码



这个阶段，是需要掌握一些计算机系统知识才能学好的，比如虚拟地址就和操作系统相关了，而函数调用栈这些又和汇编相关。

又比如很多学了很久的同学，还不太清楚变量的声明和定义区别，extern 又有什么作用，这些实际上就需要理解内存分区的东西。

C 语言的核心就在于指针、内存，能不能学好、用好 C 语言，更多在于是否拥有扎实的计算机结构、存储、运算原理方面的知识。

所以强烈建议在学习 C 语言的同时去了解一下补码、数的二进制表示、内存、汇编等知识，尤其是内存和汇编，这两个对于深刻理解指针和熟练运用有很大的帮助。

我就不仔细介绍了，要介绍的书都在这张思维导图里了。

- 深入学习指针

《C和指针》、《深入理解C指针》，真的是涉及指针方方面面的好书，墙裂推荐。



这俩兄弟懂了其实也不能帮你提高与代码的水平，主要是对于一些编译、链接过程的报错，更加的明确，不至于懵。

比如链接过程中常见的错误是符号未找到（undefined reference）和符号重定义（redefinition）

当你熟悉链接过程，符号查找过程之后，解决对应的报错也会得心应手。

- 汇编

C 语言之下就是汇编，会汇编，你就能直接把 C 语言衣服扒掉，看看背后的实现，比如大家都在讨论数组和指针有什么区别？

你去写个程序，然后 gcc -S 一下，看下汇编代码，你就会发现没啥区别。。。

汇编不需要会写，会看懂部分记得，大可不必去刻意的记住各种指令、寻址方式。

继续浏览内容

 **知乎**
发现更大的世界

打开

 **Chrome**

继续

	深入理解计算机系统中文清晰.pdf	2018 年 11 月 8 日 上午 9:58	22.3 MB	
	深入理解 C 指针.pdf	2018 年 5 月 6 日 下午 7:14	3.1 MB	PDF 文稿
	C 程序设计语言（第 2 版）.pdf	今天 下午 5:54	1 MB	PDF 文稿
	C 程序性能优化 20 个实验与达人技巧.pdf	2019 年 1 月 7 日 下午 4:37	42.4 MB	PDF 文稿
	C 和指针（第二版）.pdf	2020 年 10 月 9 日 上午 12:15	36 MB	PDF 文稿
	C 专家编程.pdf	2020 年 10 月 9 日 上午 12:14	9.2 MB	PDF 文稿

可以在这里下载，都包含在里面了：

计算机必看经典书单



mp.weixin.qq.com/s?__biz=Mzg4NjUxMz...

这一套学完，C 语言基本问题不大。

有需要的可以点赞后在评论区留言，我发给你~~

记得帮我 @编程指北 点个赞哟~~

编辑于 04-18

▲ 赞同 372 ▼

● 23 条评论

➦ 分享

★ 收藏

♥ 喜欢

收起 ^

查看全部 154 个回答