

# 轻松学通

# C语言

这是一本让你恍然大悟、大呼原来如此的国人原创基础书！

这是一本以独特方式写就的C语言程序设计书、一本C语言向导书！

左飞 李召恒 著

博客支持：<http://baimafujinji.blog.51cto.com/>

QQ群支持：134869332



## 本书特色：

- **概念讲述更通俗：**使用生动比喻和日常生活参照物、让理论有趣又易读
- **编程实践更写实：**从实际工程问题中精炼编程实例、让案例具有先导性
- **编程思想严要求：**行文力透正确程序设计思想、编码风格和程序书写规范
- **考虑学习后续性：**对数据结构和算法做铺垫、让读者能够顺畅延续知识链

中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

013071465

TP312C  
2217

# 轻松学通

# C语言

左飞 李召恒 著



TP312C  
2217



北航 C1680447

中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE



## 内 容 简 介

本书主要内容包括程序设计与C语言概述,数据及数据类型,运算符、表达式和语句,流程控制,函数,数组与字符串,指针,预处理,结构体与共用体,文件,动态数据结构等。

本书内容丰富、结构清晰、实例代码详尽,介绍经典算法、经典问题和大量的示例程序,并配有清晰插图,具有很强的参考意义。本书强调概念阐释的通俗性、编程实践的先导性、先进思想的重要性以及深入学习的后续性,可为读者更好地学习C语言编程打下坚实的基础。

本书不但适合作为C语言初学者的入门教材,也适合自学,另外也可作为大专院校相关课程的参考书以及从事C语言开发的程序员的参考手册。

### 图书在版编目(CIP)数据

轻松学通C语言 / 左飞, 李召恒著. — 北京:  
中国铁道出版社, 2013.9  
ISBN 978-7-113-17020-2  
I. ①轻… II. ①左… ②李… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第165920号

书 名: 轻松学通C语言  
作 者: 左 飞 李召恒 著

策 划: 武文斌  
责任编辑: 张 丹  
编辑助理: 吴伟丽  
责任印制: 赵星辰

读者热线电话: 010-63560056

封面设计: 多宝格

出版发行: 中国铁道出版社(北京市西城区右安门西街8号 邮政编码: 100054)  
印 刷: 北京尚品荣华印刷有限公司  
版 次: 2013年9月第1版 2013年9月第1次印刷  
开 本: 787mm×1092mm 1/16 印张: 27.75 字数: 630千  
书 号: ISBN 978-7-113-17020-2  
定 价: 55.00元

版权所有 侵权必究

凡购买铁道版的图书,如有印制质量问题,请与本社发行部联系调换。

# 21 天精通 C/C++ 的独门图文秘笈

你想学好 C/C++ 吗?

你想短时间内精通 C/C++ 吗?

21 天精通 C/C++ 不再是神话, 下面就教给你我的独门秘笈, 有图有真相!

第1-10天

自学常量、变量、数组、字符串、表达式、函数、指针.....



第11-21天

自学控制流程、结构体、引用、类、对象、多态、继承.....



第22-697天

亲手编写一些自娱自乐的小程序, 也可以借鉴一点别人的代码, 不过记住要从自己的错误中学习。



第698-3648天

和其他程序员交流。一起参与一些实际项目, 并向其他人学习。



第3649-7781天

自学高等理论物理, 建立一套时空转换的理论。



第7782-14611天

自学生物化学、分子生物学、基因工程学.....



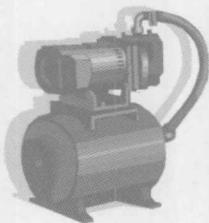
第14611天

使用你的生物知识制造返老还童的药剂。



第14611天

使用物理知识建造一台时光机器然后返回第21天。



第21天

干掉那一天的你自己!



看完我的独门秘笈是不是所有人都被“播到”了？其实这则漫画的原案来自于网络，内容虽然戏谑，但仔细品来却别有道理。

为说明这个问题，我先来谈谈所谓的 21 天效应吧。

在行为心理学中，人们把一个人的新习惯或理念的形成并得以巩固至少需要 21 天的现象，称之为 21 天效应。这是说，一个人的动作或想法，如果重复 21 天就会变成一个习惯性的动作或想法。

于是某些人开始“偷换、改造、利用”这个让人听起来貌似十分高级的概念，大炒热炒“21 天速成的神话”，于是乎一时之间——

21 天减肥、21 天塑身、21 天背会所有单词、……如此之类的东西层出不穷。

但是我想说，编程这种事要想有所修为，切不可急功近利，妄图一蹴而就！冰冻三尺非一日之寒，只有脚踏实地，从实践中取经才能学到真功夫。如果你还相信 21 天就能精通 C++ 的神话，那你还不如做梦养只机器猫呢，呵呵。

最后送大家一句话，虽然很老土，但至理名言是不怕岁月冲洗的——“书山有路勤为径，学海无涯苦作舟。”

2013 年 4 月

# 致读者

过去我在 CSDN 网站上曾经连载过《大学计算机课程学习路线》和《大学计算机课程学习路线中相关课程经典教材推介》两篇长文，有很多在校的学生一直追着读，在我的读者中反响总体来说还是很不错的。

由于是连载的文章，所以我每篇文章里都会追加上前一篇和后一篇的链接，以使读者在任何一篇文章处进入，都能顺藤摸瓜地把前前后后的内容捋清楚。然而，后来由于网站改版，虽然文章仍然被保留下来（当然是被转移到其他版块），但是原来的链接就已经作废。

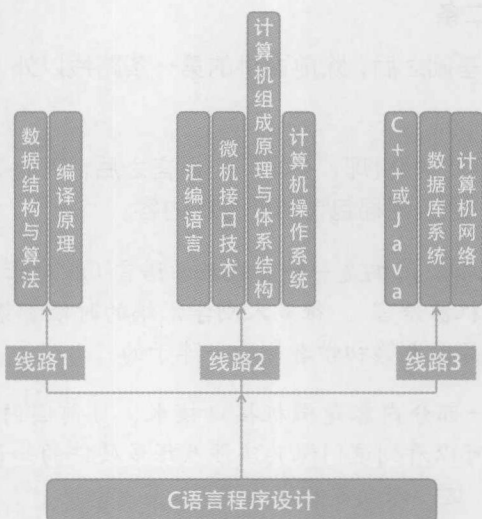
所以很多读者总是问及“为何文章中的链接无法打开？”，或者“系列中其他的文章如何能够找到”之类的问题。

这次在我的新书中，我将完整的文章奉上，并进行校正，让读者享受好作品。如果你是一个计算机相关专业的初级学生，那么这篇文章就是为你写的。

## 大学计算机课程学习路线

我之前曾经发过一篇博文《大学计算机专业那些课》来谈论大学中各门课程的意义和重要性。那是针对更广泛的范围来说的，这篇文章则更加精要和简约。因为这里研究的仅仅是核心计算机课程，更重要的是，这里要说的是这些课程之间的关系，以及它到底讲得是些什么东西。我想这些课程最终是要组成一个体系的。

请注意我这里所说的是通常情况下的核心课程，每个学校在课程设置上是可能存在差异的，但总体框架应该是不变的。



## 1. 起点：C 语言程序设计

最开始的第一门课程应该是“C 语言程序设计”，这基本上属于你日后一切学习的开始，简单地说，这门课程的主要目的和内容就是介绍一种计算机语言的语法内容，至于怎么学好它，之前的文章都谈过，这里就不再赘述。

## 2. 课程三条线：第一条

之后的课程会分为三条线，第一条线的第一门课程是“数据结构与算法”。这很简单，有一些计算机语言基础之后，为更好地使用语言去解决问题，就必须学习结构知识和算法。至于怎么学好它，我这里不想费太多的笔墨——请注意，如何学不是本文的重点。本文的重点在于告诉各位，你将来都会学些什么以及该学些什么。

“数据结构与算法”的后续课程是什么？答案应该是“编译原理”。很奇怪吧？课程就是这样设计的，其实这个路线也不是我发明的，国外大学的 CS Course 体系也是这么设计的。

编译原理说白了就是介绍编译器如何工作的，是让计算机理解计算机语言的一种方法和理论。这个工作过程之中蕴含了大量的算法，而且是非常复杂的算法，很多人甚至认为编译原理是计算机专业本科阶段最难的一门课程。

这是因为这里面的算法太复杂。当然也会用到很多数据结构，树可能是用得最多的。自然而然地，这就是相当于是数据结构与算法课程的一个后续性课程，类似一道附加题。最近听一个学生说有些学校本科已经取消这门课了，我想应该是较少一部分学校的做法吧。不过，国内部分学校的学生并不太重视这门课倒是真的。

一般课程设置就是一个学期左右的课时。但是一般的国外学校这门课都要学上整整一年，甚至两年之久。这条线路到编译这里就应该是终点了。下面我们再来介绍剩下的两条线路。

## 3. 课程三条线：第二条

在你具有一定 C 语言基础之后，除前面讲的第一条路线以外，你已经可以开始下面这条路线的学习了。

为更好地探究计算机系统的机理，在学习 C 语言之后，下面一门需要学习的课程是“汇编语言和微机接口技术”，这门课程包含两部分的内容：

- 一部分就是汇编语言，汇编是一种更底层的语言（相对于 C 语言而言），说白了它就是一种“用助记符描述的机器语言”。很多人刚学汇编的时候都觉得很复杂，因为它跟硬件紧密相关，就是那些个寄存器也够初学者糊涂一阵子的。

- 这门课程的另外一部分内容是微机接口技术，目前国内高校这部分内容都是基于 80X86 系列芯片的。已经可以看到这门课其实涉及很多硬件的知识，其实还不止这些，关于 I/O 控制、寄存器、中断，这门课里是都有体现的。



接下来要学习的内容就是“计算机组成原理与体系结构”。

其实我特别奇怪，我在写学习路线 1 的时候，有读者认为在学习 C 语言之前，或者在学习其他专业课程之前，最先要学的就是这个东西( 请注意由于原文是网站上发表的系列文章，所以可以看到读者针对具体部分的回复或留言 )。

我不知道大家是不是也觉得大学第一门专业课就应该是这个，我觉得这么想的人是不切实际、信口开河的。这门课程还是比较深的一门课程的，在汇编和接口之后安排这门课程是比较合理的。

不过，国外的一些先进的计算机教育体系确实也有在一开始的时候开设一门概论课来作为导引，有时候叫 Computer System Introduction，有时候叫 Computer Culture，其实内容差不多，作为计算机专业课程的导引课程，这些课程的设置里面已经涉及组成原理和 OS 的初步内容。但这和 Computer Structure & Architecture 还是有区别的。

接下来的一门课程是“计算机操作系统”，如果外行听起来可能觉得这个课应该是学如何使用操作系统的吧，就像“Windows 7 速成”？非也！这门课程是讲 OS 是如何设计的，或者说是如何实现一个 OS（当然实际做肯定要比说起来难多了）的。

之所以这门课程被安排在组成原理之后，那是因为这是反映一种自底向上的学习路线。操作系统当然是在计算机系统之上来搭建的。像调度啊、虚拟内存啊这些问题都是由操作系统来控制的。

我所写的《代码揭秘》一书基本上是属于这条路线上的内容的，不同的是我采用的是一种更接近读者理解方式的描述，因为在讲述这些原理之前，我都是采用 C 语言实例来作为引子的。有空也推荐大家看看，学校里的图书馆应该是可以借到的。以上就是大学计算机课程学习的第二条路线。

#### 4. 课程三条线：第三条

下面我们来介绍大学计算机课程学习的最后一条路线。

最后一条路线属于应用型的，也就是说当你学完基本的原理之后：

一方面为日后更好的工作。一方面如果从事研究工作也可择其一而继续深入探究，那么最后终究是要上升到一定应用层面的。

在你有了一定的 C 语言基础之后，学校可能会开一些面向对象程序设计方面的课程，可能是 C++ 或者 Java，因为面向对象的语言在实际开发中正广泛地被使用。无论是 C++ 还是 Java，对于 C 语言来说本质上都是语言，只不过他们多了一些 OOP 方面的特性。

接下来的一门课程应该是“数据库系统”，这门课不是教你怎么使用 SQL Server 或者 DB2 的，学校里的课程远没这么前沿。数据库系统这门课依然偏重于更多理论——尽管数据库相比于组成原理和 OS 来说已经是上层建筑了。这门课讲的大多是些数据库系统设计方面的原则，例如如何解决关系问题或者稍微涉及一部分 SQL 语言的内容等。



再之后可能会学的课程是“计算机网络”，这门课里会更加系统地向你介绍网络方面的知识，TCP、UDP、HTTP 这些名词会粉墨登场。注意这门课一般不是教你如何布网的，例如如何配置路由器这种东西一般是 CCNA 里面的内容。这门课里讲得更多的是上层原理与协议。

### 5. 三条线的关系

以上就是一般的大学计算机课程学习的几条路线。它们彼此联系，相互作用。注意我这里讲的内容不是给你推荐学习方向，而是告诉你大学里会学什么，他们之间联系如何。

从之前的文章评论里不难看出，有的同学好像根本没理解文章的意思，说自己好像更适合哪条路线的样子，其实适不适合哪条路线的问题根本不存在，这些东西你都得学，既然学就要学好。

本身就是一个体系的，怎么能做瞎子摸象的事情呢？现在网络上资源很多，想学好不怕找不到资料，但也能客观地看出浏览网络资源的人真的是分好多等的，有的是谦逊而好学的书生，有的则是蜻蜓点水的过客。

浮躁之气甚嚣尘上，胸中无物自然谈吐轻浮，看什么都不求甚解，浅尝辄止，多少有些自以为是、妄自尊大了。希望大家都能保持一颗平常心，戒骄戒躁，平和地理头学习吧！

## 大学计算机课程学习路线中相关课程经典教材推介

前阵子，连发三篇博文，向同学们介绍大学计算机课程学习路线，短短几周浏览量已经近十万。首先感谢同学们的支持，从留言中我得知，有的同学希望我在讲述课程路线的同时给出一些经典教材或参考书目以便于大家自学。

下面我就针对其中提到的一些课程给出参考书目，这些书目中外皆有，都是被广泛采用的经典教材，可供大家学习所用。

需要提醒读者注意的是，对于计算机的学习可以读的书非常之多，而这里我所推荐的书籍是根据前面规划的学习路线来的，也就是说，这里提到的书都是围绕课程学习而展开的，因此像《深入理解计算机系统》、《编码——隐匿在计算机软硬件背后的语言》这些牛书虽然非常经典但并没有辑录在此。

### 1. 给出“数据结构与算法”课程及推荐参考书目

(1) 《数据结构与问题求解 (C++版)》，Mark Allen Weiss，清华大学出版社。

这本书我觉得真是不错，算法和数据结构基本都讲到，系统、充实，例子也很好，讲得也比较容易理解。而且里面的数据结构和算法都配有完整的代码实现，绝对能运行，真是不容易啊！

老外这点真叫人感动，人家教授都是自己给自己的书编代码呀！中国的书里一般都是伪代码，呵呵，这样也省着调试。看了这个书，再做做它后面的习题，理论与编程能力双提高啊！缺点就是太厚了，不知道你有勇气读完没有。

目前中文版、英文版国内都可以买到，我中文版、英文版都看过，中文版翻译和排版错误特多，大家小心。如果英文过 CET4，直接读英文版吧，挺通俗的。

(2)《计算机算法设计与分析》，王晓东，电子工业出版社。

这是一本国人写的专门讲算法设计与分析的书，喜欢专门搞算法而且已经对数据结构很明白的同学可以看下，不过属于比较难懂得。算法就是这样的，对逻辑思维要求更高些。

(3)《算法与数据结构——C 语言描述》(第2版)，张乃孝，高等教育出版社。

这其实是一本高校教材。大家别瞠目，因为没办法，有的网友非得让我给他推荐一本教材，我也很无奈。这方面的教材我看过好几本，因为都看过，所以才有资格说说他们的情况。

我个人觉得这本书的好处是：首先是书很薄也不贵，而且内容比较规范，没有东抄西抄的现象，讲得也挺通俗易懂的。但是毕竟很薄，所以内容谈不上特别充实吧（但基本该讲的，也都讲到了）。

另外，其不足就是，里面的代码都是伪代码（就是没办法运行的，国内教材基本都是这个样子的），或者说代码很少，都是文字描述。能帮大家提高理论水平，但是学了这种书，我感觉还是不会用编程的方式来解决实际问题。

(4)《C++数据结构原理与经典问题求解》，电子工业出版社。

这个书是我写的。500 多页，有点厚。我个人感觉讲得还算通俗易懂吧，有些复杂的算法用了很大的篇幅去描述算法的过程，而且图特别多，一本书下来大概有 400 多张图表，我觉得这样更容易帮助读者理解。

另外一个特点是，这书里面的代码都是可以运行的，绝非伪代码，这里面结合具体的问题和一些经典实例都配有完整的代码。

这样能够有助于读者培养实际的问题解决和编码实现能力。但是本书的不足之处在于，印刷错误比较多——这个很让人郁闷，应该有二十几处，我自己有勘误表，你自己看的时候就得认真读啦，都是些一眼就看得出来的，总之，不会把你给带到沟里去的。

## 2. 我向读者推荐“C++程序设计”课程相关的参考书目

(1)《C++编程思想》，英文名《Thinking in C++》，Bruce Eckel，机械工业出版社。

Bruce Eckel 同样是《Thinking In Java》的作者，他的这两部作品在全世界范围内都非常畅销。《Thinking in C++》曾被美国“软件开发”杂志评选为 1996 年图书震撼大奖（Jolt Award）——一个关于技术作品的重要大奖，是对书籍内容的顶级认可。

我个人认为这本书的最大好处就在于充分体现“循序渐进”的思想，作者引导读者学习完全是经过系统的设计与思考的，是逐步引导读者的一个有效过程。这本书最初主要是从 C 中的一些内容讲起的，对于那些学过 C 的人来说，进而学习 C++，这部作品可谓不二首选。

目前该书的中英文版国内都可以买到，我强烈推荐英文版，因为中文版实在惨不忍睹，特别容易误导读者。我的博客原文上有部分错译比对，读者可以去查阅，考虑到辑入本书可能有碍观瞻，在此将其略去。

(2)《C++ Primer》，这本书名气很大，好书一本，我这里就不多说了。

同样的问题，很多人抱怨自己英文太差，看外版书看不懂，所以无奈只能去买翻译版的书。我的建议是“宁缺毋滥”，如果你没办法接受英文书，那还不如看中国人写的，毕竟有些国人的作品还是很不错的。如果你特别觉得中国人的书就是不看而去翻翻译版的书，那就是自欺欺人，要知道如果翻译得不好，你只会越看越糊涂，越学越困惑。

下面就推荐两本中国人写的作品，孔夫子曾经说过：“善哉，祁黄羊之论也！外举不避仇，内举不避子，祁黄羊可谓公矣。”意思就是说举荐贤能，要内不避亲、外不避仇。下面两本书的作者分别是钱能和白乔。

我跟钱老师素不相识，更别说有仇，大家可以放心地去看他的书，我跟白乔很熟，还合作写过一本书，这里也算是“内不避亲”吧，至于大家会不会去看书就由大家自己来决定啦！

(3)《C++程序设计教程》(第二版)。

该书是普通高等教育十一五国家级规划教材，曾获第七届全国大学出版社优秀畅销书奖，作者钱能。这本书的特点在于条理清晰，表述准确，内容丰富，贴近中国学生的阅读习惯。大家可以试试看。不过这也是标准的“教材”，如果没有耐心，恐怕会读不下去，就看你自己啦！

(4)《标准C++开发入门与编程实践》，白乔编著，电子工业出版社出版。

这本书的特点在于它是目前国内针对标准C++的第一本书，(什么是标准C++，你可以自己去网上查查看，总之是既难得又重要的东西)。另外，这本书主要是面向初学者的，要求层级不高，而且作者在表述上也非常灵活，常常引用一些像西游记里的人物和典故来带领读者学习。阅读起来非常轻松，而且不会感到枯燥，是初学者一个很好的选择。

### 3. 向大家介绍“计算机组成原理与体系结构”课程的经典书籍

(1)《计算机组织与体系结构》性能设计，英文名《Computer Organization And Architecture: Designing For Performance》，William Stallings，清华大学出版社。

这本书现在已经出到第7版，可想其读者口碑也必然极佳。该书是介绍当代计算机体系结构主流技术和最新技术的优秀教材。

作者以Intel Pentium 4 和 Power PC 作为考察实例，将当代计算机系统性能设计问题和计算机组织与体系结构的基本概念及原理紧密联系起来。而作者William Stallings教授算是国际上首屈一指的教材作家，他曾经写过许多关于计算机方面的教材，涉及范围也很广，像操作系统，组成原理和网络安全均有著述。

我个人读这本书的感觉就是内容全面翔实，且紧跟技术发展。由于计算机硬件技术是飞速发展的，而这本书自从问世以来，作者就一直不懈地对其进行更新和完善，相比之下国内的类似教材就老旧多了。不过这本书的内容非常多，开本也很厚实，如果不是很有耐性不一定读得下去。

(2) 计算机组成原理(第2版), 唐朔飞, 高等教育出版社。

该书作者是哈尔滨工业大学的唐朔飞教授。唐老师是国家级教学名师奖获得者。从教几十年, 深受学生爱戴。师德堪称典范。而这本书也是唐老师几十年教学经验的心得结晶, 我上学的时候, 我的组成原理老师对这本教材就推崇备至。

后来我在撰写《代码揭秘》时还特别从书店里买了这部书的最新版, 仔细参读之下, 不由觉得这本书条理清晰, 讲述明细, 非常容易理解, 实在不愧为国人所著的经典教程。

#### 4. 下面向大家介绍“编译原理”课程的经典书籍。

(1) 编译原理(原书第2版)

因为这本书封面画了一条龙, 所以又被称为“龙书”。这本书名气太大了, 我觉得几乎不用我介绍, 他的作者是美国的 Alfred Aho、Monica Lam、Ravi Sethi 和 Jeffrey Ullman, 目前国内版由机械工业出版社出版。

它号称是编译领域里程碑式的经典著作。不过说实在的, 编译原理这门课可是号称计算机专业本科最难的课程, 这本书对于底子不行, 理解力跟不上的人来说, 无异于天书啊! 大家自己掂量着来吧。

(2) 编译原理及实践, Louden, 机械工业出版社。

我个人感觉, 这本书要比龙书更适合初级读者。这本书的特点在于, 结合对现代编译器设计理论的详细研究和讲述, 作者自己设计了一个小型语言 Tiny, 并且为这个语言开发了完整的编译器, 书中也包含了这个编译器的完整源代码。

这就是所谓的理论与实践相结合吧, 这点非常值得推崇。读者可以一边学习, 一边跟随作者去编写那个编译器, 这样对于理论知识的理解也会更加深入。不过, 我又要说那个问题了, 这个书的翻译实在太差了。本来编译就那么复杂, 不容易理解, 如果翻译的质量又非常差, 那就真叫天书了。

#### 5. 向大家介绍“计算机网络”课程的经典书籍。

(1) 《计算机网络:自顶向下方法(第5版)》, James Kurose、Keith Ross, 机械工业出版社。

这本书是当前世界上最为流行的计算机网络教材之一, 被世界上许多大学所采用, 作者独创的自顶向下的方法来讲授计算机网络的原理及其协议, 即从应用层协议开始沿协议栈向下讲解, 强调应用层范例和应用编程接口, 学以致用, 并附有一些 Java 程序片段。



读者在掌握一定的协议和原理之后其实可以在此基础上进行一定的底层开发（如果有 Java 基础的话），作者非常注重教学的方法，深入浅出地重点讲解计算机网络的基本原理。总之这是一本不可多得的计算机网络教材。

（2）Java 网络编程精解，孙卫琴，电子工业出版社。

孙卫琴是目前国内非常有名的 Java 女作家，著述颇丰，而且其作品也非常畅销。这本书并非学校里的教材，因此对原理性东西涉及不深，它更偏重于网络开发方面的内容，书中结合大量的典型实例，详细介绍用 Java 来编写网络应用程序的技术。

我个人觉得作为前面那本讲解原理的书籍的补充和课余读物是非常合适的。有兴趣的同学可以参考一下，但是要注意，你必须有一定 Java 基础，否则你是不适合阅读这本书的！

（3）思科网络技术学院教程，人民邮电出版社。

前面的书籍都是偏重于软件的，但是在实际工作中，一些网络工程师可能需要一定的操作网络设备的能力，这其实更侧重于硬件。注意这已经不同于开发工作。这时你可以尝试参考一下思科网络技术学院教程。

准确地说，这其实是一套书籍的总称，Cisco 的书有一大堆。面向 Cisco 的各种级别和种类的认证，从 CCNA 开始，CCNP 直到 CCIE，逐层深入。因为本身在通信公司，所以身边有很多 CCIE 认证专家，对这些东西耳濡目染，尽管自己不算是这个方向的，但据那些 CCIE 牛人讲这套东西还是很有用的，那好处可是多了去了啊！

对于普通的本科同学来说，CCNA 的内容已经足够了，CCNA 的教材一共两本，四个学期，里面有大量的七层模型，TCP、UDP、路由器配置方面的知识，非常实用。

## 6. 向大家介绍“计算机操作系统”课程的经典书籍

操作系统方面的好书实在太多了，我这里没办法尽诉，就举三个侧重点各不相同，而且各具特色的给大家吧！

（1）操作系统——精髓与设计原理（第 5 版），William Stallings，电子工业出版社。

William Stallings 前面的文章中已经出现过了，他是目前国际上非常著名的计算机教材作者，他的许多教材被全世界范围内的众多大学所采纳。这本著作第四版曾获得美国计算机科学与工程类教材大奖，得到美国教育界和工程技术人员的好评。

本书不仅全面系统地讲述了操作系统的基本概念，且以当代最流行的操作系统——Windows、UNIX 和 Linux 为例，全面清楚地展现了当代操作系统的本质和特点，使本书的内容具有先进性和适应性。是操作系统方向的首选经典教材。

（2）《UNIX 环境高级编程（第 2 版）》，Richard Stevens、Stephen Rago，人民邮电出版社。

本书被誉为 UNIX 编程“圣经”，对于所有层次 UNIX/Linux 程序员都是一本不可或缺的参

考书。要提醒读者的是，虽然名为 UNIX 编程，但本书绝大部分内容也同样适用于 Linux。而且两者不同之处，作者也进行了特别说明。但是，这本书跟前面那本不同，它不是讲操作系统实现的，而是讲开发的。

书中介绍的内容大致包括 UNIX 文件和目录、标准 I/O 库、系统数据文件和信息、进程环境、进程控制、进程关系、信号、线程、线程控制、守护进程、各种 I/O、进程间通信、网络 IPC、伪终端等方面的内容，还在此基础上介绍了多个应用实例，包括如何创建数据库函数库以及如何与网络打印机通信等。

这本书太经典了，总之一句话，如果日后要从事 UNIX/Linux 相关的开发工作，这本书应该是必备的！

(3) Orange'S: 一个操作系统的实现，于渊，电子工业出版社。

这是一本畅销书的升级版。作者一步一步地向读者呈现一个操作系统框架的完成过程。书中不仅关注代码本身，同时关注完成这些代码的思路和过程。

本书不同于其他的理论型书籍，而是提供给读者一个动手实践的路线图。这正是本书的最大特点，它其实是带着你一起去亲手编写一个操作系统，这实在是一件不可思议的事情。这是中国人写的一本非常优秀而且畅销的作品。不过我也要提醒一下读者，阅读这本书的人要有一定的理论基础，否则将很难理解其中的思路和内涵。

### 是什么原因使我们学不好编程？

之前我在网上最常被问到的问题之一，可能是像这样的：

老师，请问我是应该学 C++ 还是学 Java 呢？

老师，听说 C 语言已经不行了，.net 才是未来的方向，是不是这样的？

老师，我原来学过 C，现在想学 C#，但是好像听说 Java 比较好找工作，请问我该怎么选择？

其实很多人这样犹疑不定，想学 C 又觉得可能学 C 没有学 Java 有前途，想学 Java 好像又觉得 Java 好像比 C++ 难学（而且他其实连 C++ 和 Java 是怎么回事都没搞清楚）最后都是一事无成，C 没学完，Java 也没学明白，C++ 也是个半吊子。

与其在这里思前想后，左右为难，到不如痛下决心，一路到底。很多人大学四年基本上都在这样的犹犹豫豫中度过，而其实要把《Thinking in Java》通读三遍可能也只需要一个学期的时间。

花时间来郁闷、犹豫、抉择、困惑，就是“等待成本”。花时间来体验、尝试、实践，就是“穿越成本”。当一个人等待与拖延的成本远远高于他真正开始行动所需要的成本，他就会慢慢陷入越等待越不行动的怪圈。我把这个模式称为“等死模式”。

与其在等死模式中消耗自己的心力与体力，还不如去试一试！其实在某些时候没选择反而就是最好的选择。

## 1. 沈绪榜院士的中肯答案

我在上学的时候有一次，中国机载计算机奠基人之一的沈绪榜院士为我们做报告，报告之后进入自由问答阶段。一位正值大四的学生起身问道，沈院士，请问您当初为什么会选择计算机作为自己的专业方向？

院士笑言道，其实在进入这个领域之前，我们根本连计算机长什么样子都没有见过。当时就是服从国家分配，国家让我去学计算机我就去学。当时我们甚至连二进制都还没有概念，华罗庚是我们的老师，他就从二进制给我们讲起……

因为没有选择，也就没有犹豫，无论是国家分配，还是命运使然，看似误打误撞进入这个领域的当年的那个同我们一样的学生，现在已经成为一名院士。

从这个角度看，是否印证了前面的论调呢？正是因为选择太多，人们才会更加迷茫，而迷茫的人最容易进入等死模式。所以那些整天还在问我到底是该选择工作、还是选择考研；到底是应该走 Java 方向，还是应该走 C++ 方向的学生都有进入等死模式的前兆。

说白了，这种等待成本将远远大于穿越成本。一旦你陷入了等死模式，最好的选择就是行动起来，进入穿越模式！穿越也许会有短期痛苦，但是等死往往会带来更大的永久损失。然而，强者毕竟是少数。

很多人即使已经意识到问题的严重性，他也很难做出实质性的作为。因为任何成功都不可能是一蹴而就的。如果你眼前有一座很高的墙，而你想翻越它，那么这时我会建议你应该先把一只鞋扔过墙那边去！这已经不是什么新鲜的论调了。

《史记·项羽本纪》中记载：“项羽已杀卿子冠军，威震楚国，名闻诸侯。乃遣当阳春、蒲将军将卒二万渡河，救钜鹿。战少利，陈馥复请兵。项羽乃悉引兵渡河，皆沉船，破釜甑，烧庐舍，持三日粮，以示士卒必死，无一还心。”“破釜沉舟”的故事正是现实中一个鲜活的例子。正因为“破釜沉舟”，所以才“别无选择”，因此才能“孤注一掷”。

问题的关键就在于勇敢地迈出第一步。这时，可能知易行难的窘境又再次出现。很多人说要下定决心学好 C 语言，就满腔热情地请老师给推荐几本书，这种推荐仅我一人已经做过很多。

## 2. 切忌夸夸其谈和好高骛远

很多学生看老师的推荐书单后都表示感谢和欢迎。但当初那些让老师推荐书单的学生，是不是都去读了书单上的书呢？哪怕一本？事实是没几个人的。这种三分钟热情的人基本上很难有所作为，特别是想学习编程这种需要多多实践的东西，那些夸夸其谈和好高骛远的人都只会被拉得更远。

正在读本文的你不妨自己思量一下现在的自己到底处在哪种状态？

《尚书》中曰：“说拜稽首曰：‘非知之艰，行之惟艰。’”孔传：“言知之易，行之难。”这就是“知易行难”的出处。很多事情都是这样的，过去我在网上经常会发表一些个人的看法，同学们也会发表一些自己的感想。于是紧接着便会有很多学生前仆后继地留言，一方面恍然大悟地发现原来应是如此，一方面也悔不该当初误入歧途。

然后呢？其实很多道理已经被反复讲过千百遍了，即使我们不是，你也应该知道。但就是“知易行难”啊。大家都明白是非曲直，但能做到的人寥寥无几。

很多学生抱怨数据结构和算法很难不容易学，很多人觉得C语言总是入不了门，有人觉得自己耽误那么多时间现在恍然大悟发现自己大四该找工作又突然不知所措……于是左一个拜师，右一个求教。左让老师推荐一本书，右让老师提点一下学习方法。大家话也讲了，招也支了，但是请同学们扪心自问自己实际做得如何。知易行难啊！

从这个角度来看我前面的很多文章就统统变成精神鸦片。当你昏昏欲睡的时候，就来这打一针提提神。然后呢？神清气爽，仍然无所事事。该看的书还是没看，该做的事还是没做。要知道只看老师们发的这些东西是不能学好编程的，关键是人家推荐的书要仔细读，书后面的练习题要认真做。

如果你真的读了老师推荐的书，自己动手做了练习，那样才能内化出自己的感悟，这样再和老师交流才能碰出新的火花。不然跟老师一对上你就自动熄火了。其实如果是这样的话，我真是觉得就没必要在这浪费时间了，还不如下载个电影看看算了，自己同样觉得开心。

每个人的层次不可能完全统一，总是有长有短。这是很正常的。但是，在这里我又要刺痛同学们了。态度决定一切，关键要看你是抱着一种什么样的态度来学习。有的人真是抱着“三人行必有我师”的谦虚的心去向别人求教，以期自我提升。

而自以为是的人一直都是存在的，笛卡儿说我越是学习就越是认识到自己的无知，如果你满以为自己已经很强了，足以藐视这里所有的专家，那只能说明你已经浅薄得不堪入目。

浮皮潦草，浅尝辄止的人就更多了，说实在的有些老师发技术文章可能稍长了些，试问点进去的朋友又有几个能从头看到尾？很多人都是蜻蜓点水而已。所以往往会提一些似是而非的问题，发一些风马牛不相及的留言。

知难行易，知易行难，知先行后，行先知后，众说纷纭。要知道这个古老的哲学命题引起了很多人的关注。理学家朱熹也曾提出过“知先行后”的说法。后来，到了明代，心学的大师王阳明又创造性地提出了新的观点，王阳明首先系统地提出了“知行合一”的哲学思想。

由此，这一在中国思想史上引起轩然大波的认识论命题，第一次提出“知”和“行”是统一的。王阳明认为，“知是行之始，行是知之成”。我这里就是借由王阳明的思想展开的一些看法。希望同学们努力克服“知易行难”的障碍，争取做到“知行合一”，好好利用各种资源来努力提高自己，不要做整天比比划划，却满腹空空的人。



# 前言

## 为什么还学 C 语言

C 语言不仅是理工科学生必修的课程，同时也是国家计算机等级考试的必考科目。时至今日，仍然有许多软件系统采用 C 语言来作为开发语言。C 语言具有语法简洁、灵活方便，支持底层操作、开发自由度大，可移植性好、执行效率高等诸多优点。

学好 C 语言不仅能够为程序设计本身打开一扇天窗，更能够帮助初学者实现举一反三、触类旁通的愿望。以 C 语言为基础纵向可以继续学习数据结构与算法方面的相关知识，横向可以为继续学习面向对象的 C++ 或 Java 语言奠定基础。可见 C 语言不失为计算机程序设计入门的首选语言。

## 初学者的困惑

尽管学习 C 语言是一件令人向往的事情，但是很多初学者在深入学习之后往往感到困惑：

- ☑ 对新的概念感到陌生，理解起来不够透彻；
- ☑ 面对复杂难懂的语法规则往往是学过之后，印象不深，写起代码又感到不知所措，无从下手。

为了帮助读者全面系统地掌握 C 程序设计的要领，帮助那些在门口徘徊却迟迟无法进入状态的读者扫清障碍、冲关破险，我们编写了这本《轻松学通 C 语言》。在这本书里笔者总结了以往学习和实践 C 程序设计的经验，运用轻松的笔调和有趣的实例引领读者步步深入，探索 C 语言的奥秘。

## 本书特点

目前，市场上可见的 C 语言著作可谓是琳琅满目、层出不穷。其中也不乏经典之作，然而时过境迁，很多写于 20 世纪八九十年代的 C 语言书籍已经跟不上快速变换的时代步伐。为了适应新一代 C 程序爱好者的需求，我们重拾了 C 语言这个传统的话题，希望能够写出一些更加实用、有特色的内容。

C 语言作为一种“活”的语言，它在不断充实和进步，我们希望本书能够不断适应时代的变换和读者的需求。为此，本书力求突出以下几个特点：