

<<一站式学习C编程>>

图书基本信息

书名：<<一站式学习C编程>>

13位ISBN编号：9787121129827

10位ISBN编号：7121129825

出版时间：2011-3

出版时间：电子工业出版社

作者：宋劲杉

页数：471

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

本书最初是为北京亚嵌教育研究中心的嵌入式Linux系统工程师就业班课程量身定做的教材之一。该课程是为期四个月的全日制职业培训，要求学员毕业时具备非常Solid的C编程能力，能熟练地使用Linux系统，同时对计算机体系结构与指令集、操作系统原理和设备驱动程序都有较深入的了解。然而学员入学时的水平是非常初级而且参差不齐的：学历有专科、本科也有研究生，专业有和计算机相关的也有很不相关的（例如会计专业），以前从事的职业有和技术相关的也有完全不相关的（例如HR），年龄从二十出头到三十五六岁的都有。

这么多背景完全不同、基础完全不同、思维习惯和理解能力完全不同的人来听同一堂课，大家都迫切希望学会嵌入式开发技术，投身IT行业，这就是职业教育的特点，也是我编本书时需要考虑的主要问题。

学习编程绝不是一件简单的事，尤其是对于零基础的初学者来说。

大学的计算机专业有四年时间从零基础开始培养一个人，微积分、线性代数、随机数学、离散数学、组合数学、自动机、编译原理、操作系统、计算机组成原理等一堆基础课，再加上C/C++、Java、数据库、网络、软件工程、计算机图形学等一堆专业课，最后培养出一个能找到工作的学生。

很遗憾这最后一条很多学校没有做好，来亚嵌培训的很多学生这四年就是这么学过来的，但据我们考查他们的基础几乎为零，我不知道为什么。

与之形成鲜明对比的是，只给我们四个月的时间，同样要求从零基础开始，最后培养出一个能找到工作的学生，而且还要保证他找到工作，这就是职业教育的特点。

为什么我说“只给我们四个月的时间”？

我们倒是想教四年呢，但学时的长短我们做不了主，是由市场规律决定的。

四年的任务要求四个月做好，要怎么完成这样一个几乎不可能的任务呢？

有些职业教育给出的答案是“实用主义”，打出了“有用就学，没有用就不学”的口号，大肆贬低说大学里教的基础课都是过时的、无用的，只有他们教的技术才是实用的，这种炒作很不好，我认为大学里教的每一门课都是非常有用的，基础知识在任何时候都不会过时，倒是那些时髦的“实用技术”有可能很快就过时了。

四年的任务怎么能用四个月做好？

我们给出的答案是“优化”。

现在大学里安排的课程体系最大的缺点就是根本不考虑优化。

每个过来人都会有这样的感觉：大一大二学了好多数学课，却不知道都是干什么用的，也不懂为什么要学。

连它有什么用都不知道怎么能有兴趣学好呢？

然后到大三大四学专业课时，用到以前的知识了，才发现以前学的数学是多么有用，然而早就忘得一干二净了，考完试都还给老师了，回头重新学吧，这时候发现很多东西以前根本没学明白，现在才真的学明白了，那么前两年的时间岂不是都浪费了？

大学里的课程体系还有一个缺点就是不灵活，每门课必须占一个学期，必须由一个老师教，不同课程的老师之间没有任何沟通和衔接，其实这些课程之间是相互依赖的，把它们强行拆开是不符合人的认知规律的。

比如我刚上大学的时候，大一上半学期就被逼着学C语言，其实C语言是一门很难的编程语言，不懂编译原理、操作系统和计算机体系结构根本不可能学明白，那半个学期自然就浪费掉了。

当时几乎所有学校的计算机相关专业都是这样，大一上来就学C语言，有的学校更疯狂，上来就学C++，导致大多数学生都以为自己会C语言，但其实都是半吊子水平，到真正写代码的时候经常为一个Bug搞得焦头烂额，却没有机会再系统地学一遍C语言，因为在学校看来，C语言课早在大一就给你“上完了”，就像一顿饭已经吃完了，不管你吃饱没吃饱，不会再让你重吃一遍了。

显而易见，如果要认真地对这些课程做优化，的确是有很多水分可以挤的。

本书有什么特点 本书不是孤立地讲C语言，而是和编译原理、操作系统、计算机体系结构结合起来讲。

<<一站式学习C编程>>

或者说，本书的内容只是以C语言为载体，真正讲的是计算机的原理和程序的原理。

强调基本概念和基本原理，在编排顺序上非常重视概念之间的依赖关系，每次引入一个新的概念，只依赖于前面章节已经讲过的概念，而绝不会依赖后面章节要讲的概念。

有些地方为了叙述得完整，也会引用后面要讲的内容，比如说“有关我们到第章再仔细讲解”，凡是这种引用都不是必要的依赖，可以当它不存在，只管继续往下看就行了。

尽量做到每个知识点直到要用的时候才引入。

过早引入一个知识点，讲完了又不用它，读者很快就会遗忘，这是不符合认知规律的。

本书面向什么样的读者 这是一本从零基础开始学习编程的书，不要求读者有任何编程经验，但读者至少需要具备以下素质：熟悉Linux系统的基本操作。

如果不具备这一点，请先参考其他教材学习Linux系统的基本操作，熟练之后再学习本书，《鸟哥的Linux私房菜》据说是Linux系统管理和应用方面比较好的一本书。

但学习本书并不需要会很多系统管理技术，只要会用基本命令，会自己安装系统和软件包就足够了。

具有高中毕业的数学水平。

本书会用到高中的数学知识。

事实上，如果不具有高中毕业的数学水平，也不必考虑做程序员了。

但并不是说只要具有高中毕业的数学水平就足够做程序员了，只能说看这本书应该没有问题，数学是程序员最重要的修养，计算机科学其实就是数学的一个分支，如果你的数学功底很差，日后还需恶补一下。

具有高中毕业的英文水平。

理由同上。

对计算机的原理和本质深感兴趣，不是为就业而学习，不是为拿高薪而学习，而是真的感兴趣，想把一切来龙去脉搞得清清楚楚而学习。

我认为程序员有三种境界 [1]1：第一种是把编程当作谋生手段的程序员，这种人学点技术找到工作就觉得修成正果了，上班时能偷懒则偷懒，一听说加班就怨声载道，下班后绝不碰跟工作相关的东西；第二种是希望通过编程不断提高自己生活品质的程序员，这种人向来认为写代码和看书学习是自己的事，是为提高自己的竞争力而做的，只不过顺便帮老板创造了价值，所以加班什么的从来不计较，不仅积极完成工作，而且每时每刻都在想如何把工作完成得更漂亮；最高境界的是把编程当作毕生爱好的程序员，这种人通常也具备第二种人的品质，此外还有一些独特之处，无论是工作中用到的技术还是与工作无关的技术都同样有兴趣钻研，无论是有报酬的工作还是无报酬的开源软件都同样积极参与，甚至退休之后仍把编程当作消遣，换句话说，编程就是一种生活方式。

我自己大概属于第二种人吧。

如果你不幸属于第一种人，那么你不适合做程序员，请放弃学习本书。

勤于思考。

本书尽最大努力理清概念之间的依赖关系，力求一站式学习，读者不需要为了找一个概念的定义去翻其他书，也不需要为了搞清楚一个概念在本书中前后一通乱翻，只需从前到后按顺序学习即可。

但一站式学习并不等于傻瓜式学习，有些章节有一定的难度，需要积极思考才能领会。

本书可以替你节省时间，但不能替你思考，不要指望像看小说一样走马观花看一遍就能学会。

为什么要学这本书而不是K&R K&R是公认的世界上最经典的C语言教程，这点毫无疑问。

在C标准出台之前，K&R第一版就是事实上的C标准。

C89标准出台之后，K&R跟着标准推出了第二版，可惜此后就没有更新过了，所以不能反映C89之后C语言的发展以及最新的C99标准，本书在这方面做了很多补充。

上面我说过了，这本书与其说是讲C语言，不如说是以C语言为载体讲计算机和操作系统的原理，而K&R只是为了讲C语言而讲C语言，侧重点不同，内容编排也很不相同。

K&R写得非常好，代码和语言都非常简洁，但很可惜，只有会C语言的人才懂得欣赏它，K&R是非常不适合入门学习的，尤其不适合零基础的学生入门学习。

本书“是什么”和“不是什么” 本书包括两大部分：C语言入门。

介绍基本的C语法，帮助没有任何编程经验的读者理解什么是程序，怎么写程序，培养程序员的思维

<<一站式学习C编程>>

习惯，找到编程的感觉。

其中前6章改编自参考文献[1]。

C语言本质。

结合计算机和操作系统的原理讲解C程序是怎么编译、链接、运行的，同时全面介绍C的语法。

第 15.1 节改编自林小竹老师的讲义，第 25.1 节和第 25.2 节改编自朱仲涛老师的讲义，第 17 章改编自参考文献[2]。

这本书定位在入门级，虽然内容很多，但不是一本百科全书，除了把C语言基本讲透了之外其他内容都不深入，书中列出了很多参考文献，是读者进一步学习的起点。

K & R的第1章是一个Whirlwind Tour，把全书的内容简单过一遍，然后再逐个深入讲解。

本书也可以看作是计算机专业课程体系的一个Whirlwind Tour，学完之后读者有了一个全局观，再去学习本书所列的参考文献就应该很容易上手了。

有的评论者批评我这本书是“满汉全席（暗含的意思就是这一本就行了）”，什么都想讲，什么都讲不细，不符合UNIX的哲学（Do one thing, do it well），我必须反驳一下了。设想有一个菜鸟，对学习计算机技术非常有热情，却不知从何处入手，于是去请教一个牛人，这个牛人就是批评我的那个评论者，以他的风格一定会给这个菜鸟推荐一个书单子（或许和本书的参考书目差不多吧），然后对菜鸟说：Study each book, study it well.如果这个菜鸟来请教我，我会首先推荐他看这本书，然后对他说：看完这本书只是一个开始，以后的路还长着呢，但我相信你看完这本书就知道以后该看什么书，而不需要再问别人了。

为什么要在Linux平台上学C语言？

用Windows学C语言不好吗？

用Windows还真的是学不好C语言。

C语言是一种面向底层的编程语言，要写好C程序，必须对操作系统的工作原理非常清楚，因为操作系统也是用C写的，我们用C写应用程序直接使用操作系统提供的接口。

Linux是一种开源的操作系统，你有任何疑问都可以从源代码和文档中找到答案，即使你看不懂源代码也找不到文档，也很容易找个高手教你，各种邮件列表、新闻组和论坛上从来都不缺乐于助人的高手；而Windows是一种封闭的操作系统，除了微软的员工别人都看不到它的源代码，只能通过文档去猜测它的工作原理，更糟糕的是，微软向来喜欢藏着掖着，好用的功能留着自己用，而不会写到文档里公开。

Windows平台上的开发工具往往和各种集成开发环境（Integrated Development Environment, IDE）绑在一起，例如Visual Studio、Eclipse等。

使用IDE确实很便捷，但IDE对于初学者绝对不是好东西。

微软喜欢宣扬傻瓜式编程的理念，告诉你用鼠标拖几个控件然后单击一个按钮就可以编译出程序来，但是真正有用的程序有哪个是这么拖出来的？

很多从Windows平台入门学编程的人，编了好几年程序，还是只知道编完程序单击一个按钮就可以跑了，把几个源文件拖到一个项目里就可以编译到一起了，如果有更复杂的需求他们就傻眼了，因为他们脑子里只有按钮、菜单的概念，根本没有编译器、链接器、Makefile的概念，甚至连命令行都没用过，然而这些都是初学编程就应该建立起来的基本概念。

另一方面，编译器、链接器和C语言的语法有密切的关系，不了解编译器、链接器的工作原理就不可能真正掌握C的语法。

所以，IDE并没有帮助你学习，反而阻碍了你学习，本来要学好C编程只要把语法和编译命令学会就行了，现在有了IDE，除了学会语法和编译命令，你还得弄清楚编译命令和IDE是怎么集成的，这才算学明白了，本来就很复杂的学习任务被IDE搞得更加复杂了。

Linux用户的使用习惯从来都是以敲命令为主，以鼠标操作为辅，从学编程的第一天起就要敲命令编译程序，等到你把这些基本概念都搞清楚后，你觉得哪个IDE好用你再去用，不过到那时候你可能会更喜欢vi或emacs而不是IDE了。

有的读者目前只会用Windows系统，但想学习C编程，能否从本书学起？

为了照顾只会用Windows系统的读者，我也会介绍Windows下的C开发环境，本书绝大多数程序只

使用标准C的语法和库函数，在Windows上也可以编译运行，但是有关调试、操作系统原理和链接原理的章节则完全以Linux系统为背景来讲解，Windows平台上的链接和调试相关知识可以参考电子工业出版社出版的《软件调试》和《程序员的自我修养——链接、装载与库》。

总之，除了第10章、第17章、第18章、第19章、第21章之外，本书其他章节都可以在Windows平台上学习。

本书是《Linux C编程一站式学习》的升级版，它和前一版有什么不同？

自《Linux C编程一站式学习》出版以来，我收到了很多读者反馈，让我意识到很多地方写得还不到位，所以决定修订和再版。

我认为写一本书应该是一个持续改进的过程，一旦停笔，这本书也就失去了生命力。

所以这次修订也不会是最终版本，只要还有不十分满意的地方，日后我还会继续修订下去。

本书相对于前一版的改动仅限于局部细节，已经买过前一版的读者朋友可以不必购买本书。

很多读者期望本书网络版中的Linux系统编程部分也能修订出版，很遗憾的是目前我还没有精力修订那一部分。

下一步计划可能会把Linux系统编程部分单独成书出版，但请不要抱有期望，写书毕竟不是我生活的主线，只是业余的、兴趣驱动的和半公益性质的活动，请读者朋友们谅解。

本书相对于前一版的改进主要有以下几方面。

本书是一本C语言教材而不是Linux教材，事实上绝大部分章节是平台无关的，上一版过于强调Linux平台，使很多Windows平台的初学者错过了本书。

我已经知错了，特将书名中的Linux字眼去掉，并且在书中补充介绍了Windows下的C开发环境。

本书是一本C语言教材而不是嵌入式开发教材，一些培训公司的市场宣传给很多初学者植入了错误的观念，比如“只有嵌入式开发才用C语言，做别的开发不需要学C语言”。

本书的上一版很不幸地归类在“嵌入式技术丛书”里，为避免强化嵌入式和C语言之间的错误关联，也为了避免读者误以为本书是培训公司用剪刀浆糊拼凑出来的Yet another嵌入式教材，本书不再归类于“嵌入式技术丛书”。

虽然我在上一版中信誓旦旦地说这是一本面向完全零基础读者的书，但现实教育了我，要写出一本让任何零基础读者都看得懂的书是一门复杂的系统工程，我只能努力接近这个目标，而永远达不到这个目标。

感谢我老婆桃子和我表弟大鹏，他们作为真正的零基础读者仔细Review了我的书，尤其桃子还是一个文科生，她每次给我指出的问题都完全出乎我的意料——原来还有人会这么理解啊，原来还有人用这样的思维方式啊。

经过一番修订之后，现在我还是要说这是一本面向完全零基础读者的书，至少我一直是朝这个目标努力的。

修正了读者指出的一些小错误和不恰当的表达，有些地方阐释得不够我又加了些笔墨，有些技术细节改用更好的实现方式，比如重写了第21.4节，我认为现在用的方法比上一版要好。

内容概要

本书有两条线索，一条线索是以Linux平台为载体全面深入地介绍C语言的语法和程序的工作原理，另一条线索是介绍程序设计的基本思想和开发调试方法。

本书分为两部分：

第一部分讲解编程语言和程序设计的基本思想方法，让读者从概念上认识C语言；第二部分结合操作系统和体系结构的知识讲解程序的工作原理，让读者从本质上认识C语言。

本书适合做零基础的初学者学习C语言的第一本教材，帮助读者打下牢固的基础。有一定的编程经验但知识体系不够完整的读者也可以对照本书查缺补漏，从而深入地理解程序的工作原理。

本书最初是为北京亚嵌教育研究中心的嵌入式Linux系统工程师就业班课程量身定做的教材之一，也适合作为高等院校程序设计基础课程的教材。

本书对于C语言的语法介绍得非常全面，对C99标准做了很多解读，因此也可以作为一本精简的C语言语法参考书。

作者简介

宋劲杉，亚嵌教育资深讲师，清华大学自动化系硕士，6年嵌入式系统开发经验，3年嵌入式行业教学经验，精通Linux内核、POSIX、TCP / IP，擅长ARM平台的Linux系统移植和应用开发，目前关注的方向有分布式系统、动态语言。

爱好：开源软件、电子音乐、HomeParty。

书籍目录

上篇 C语言入门

第1章 程序的基本概念

- 1.1 程序和编程语言
- 1.2 自然语言和形式语言
- 1.3 程序的调试
- 1.4 第一个程序

第2章 常量、变量和表达式

- 2.1 继续Hello World
- 2.2 常量
- 2.3 变量
- 2.4 赋值
- 2.5 表达式
- 2.6 字符类型与字符编码

第3章 简单函数

- 3.1 数学函数
- 3.2 自定义函数
- 3.3 形参和实参
- 3.4 全局变量、局部变量和作用域

第4章 分支语句

- 4.1 if语句
- 4.2 if/else语句
- 4.3 布尔代数
- 4.4 switch语句

第5章 深入理解函数

- 5.1 return语句
- 5.2 增量式开发
- 5.3 递归

第6章 循环语句

- 6.1 while语句
- 6.2 do/while语句
- 6.3 for语句
- 6.4 break和continue语句
- 6.5 嵌套循环
- 6.6 goto语句和标号

第7章 结构体

- 7.1 复合类型与结构体
- 7.2 数据抽象
- 7.3 数据类型标志
- 7.4 嵌套结构体

第8章 数组

- 8.1 数组的基本概念
- 8.2 数组应用实例：统计随机数
- 8.3 数组应用实例：直方图
- 8.4 字符串
- 8.5 多维数组

<<一站式学习C编程>>

第9章 编码风格

- 9.1 缩进和空白
- 9.2 注释
- 9.3 标识符命名
- 9.4 函数
- 9.5 indent工具

第10章 gdb

- 10.1 单步执行和跟踪函数调用
- 10.2 断点
- 10.3 观察点
- 10.4 段错误

第11章 排序与查找

- 11.1 算法的概念
- 11.2 插入排序
- 11.3 算法的时间复杂度分析
- 11.4 归并排序
- 11.5 线性查找
- 11.6 折半查找

第12章 栈与队列

- 12.1 数据结构的概念
- 12.2 堆栈?
- 12.3 深度优先搜索
- 12.4 队列与广度优先搜索
- 12.5 环形队列

本阶段总结

下篇 C语言本质

附录A 字符编码

参考文献

索引

编辑推荐

畅销书升级版 手把手教你将四年的任务用四个月做好 以C语言为载体，系统讲述计算机原理和程序原理 从零基础开始学习编程，内容涵盖C语言入门及C语言本质

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>