第一图书网, tushu007.com

<<Linux C编程一站式学习>>

图书基本信息

书名：<<Linux C编程一站式学习>>

13位ISBN编号：9787121097713

10位ISBN编号：7121097710

出版时间：2009-12

出版时间：电子工业出版社

作者：宋劲杉

页数：461

字数：690000

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语言编程能力，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比较深入的了解。
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）；年龄从二十岁出头到三十五六岁的都有。
这么多背景、基础、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，也是我编写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、概率论、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、网络工程、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，据我们考查，来亚嵌培训的很多学生基础几乎为零，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好工作，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”有可能很快就会过时了。
四年的任务怎么才能用四个月做好？
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，不明白为什么要学。
连它有什么用都不知道怎么能有兴趣学好呢？
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，才发现很多东西以前根本没学明白，现在真的学明白了，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，每门课必须占用一个学期，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习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，大一刚来就学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进行优化，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。
或者说，本书的内容只是以C语言为载体，真正讲的是计算机和程序的原理。
强调基本概念和基本原理，在编排顺序上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，而绝不会依赖于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，只管继续往下学习就行了。
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相关知识，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，日后还需要恶补一下。
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。
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阅其他书籍，也不需要为了搞清楚一个概念在本书中乱翻一通，只需要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，需要读者积极思考才能领会。
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　　《The C Programming Language》（后文简称[K&R]）是公认的世界上最经典的C语言教程之一，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。
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以及怎么写程序，培养程序员的思维习惯，找到编程的感觉。
前半部分改编自《How To Think Like A Computer Scientist: Learning with C++》（后文简称[ThinkCpp]）。
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。
位运算的章节改编自林小竹老师的讲义；链表和二叉树的章节改编自朱仲涛老师的讲义；汇编语言的

章节改编自《Programming from the Ground Up: An Introduction to Programming using Linux Assembly Language》（后文简称[GroundUp]），在该书的最后一章中提到，学习编程有两种Approach，一种是“Bottom Up”，一种是“Top Down”，它们各有优缺点，而我们需要将两者结合起来。
所以我编写本书的思路是：第一部分Top Down；第二部分Bottom Up；第三部分可以算填补了中间的空隙，三部分全都围绕C语言展开。

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C语言的基础知识要讲透之外其他内容都不深入，书中列出了很多参考资料，是读者进一步学习的起点。
[K&R]的第1章是一个Whirlwind Tour，把全书的内容简单概括了一遍，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，学习完本书之后读者有了一个全局观，再去学习那些参考资料就应该很容易上手了。

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，从而更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 第2章 常量、变量和表达式 第4章 分支语句 第5章 深入理解函数 第6章 循环语句 第7章 结构体 第8章 数组 第9章 编码风格 第10章 gdb 第排序与查找 第12章 栈与队列 下篇 C语言本质 第13章 计算机中数的表示 第14章 数据类型详解 第15章 运算符详解 第16章 计算机体系结构基础 第17章 x86汇编程序基础 第18章 汇编与C之间的关系 第19章 链接详解 第20章 预处理 第21章 Makefile基础 第22章 指针 第23章 函数接 第24章 C标准库 第25章 链表、二叉树和哈希表 附录A 字符编码 索引 参考书目

章节摘录

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　程序（Program）告诉计算机应该如何完成一个计算任务，这里的计算可以是数学运算（如解方程），也可以是符号运算（如查找和替换文档中的某个单词）。
从根本上说，计算机是由数字电路组成的运算机器，只能对数字进行运算，程序之所以能进行符号运算，是因为符号在计算机内部也是用数字表示的。
此外，程序还可以处理声音和图像。
声音和图像在计算机内部必然也是用数字表示的，这些数字经过专门的硬件设备转换成人可以听到的声音和看到的图像。
程序由一系列指令（Instruction）组成，指令是指示计算机进行某种运算的命令，　通常包括以下几类：　输入（Input）　从键盘、文件或者其他设备获取数据。
输出（Output）　把数据显示到屏幕，或者存入一个文件，或者发送到其他设备。
基本运算　执行最基本的数学运算（加减乘除）和数据存取。
测试和分支　测试某个条件，然后根据不同的测试结果执行不同的后续指令。
循环　重复执行一系列操作。
对于程序来说，有上面这几类指令就足够了。
你曾用过的任何一个程序，不管它有多么复杂，都是由这几类指令组成的。
程序是那么复杂，而编写程序可以用的指令却只有这么简单的几种，这中间巨大的落差就要由程序员去填补了，所以编写程序理应是一件相当复杂的工作。
编写程序可以说就是这样一个过程：把复杂的任务分解成子任务，把子任务再分解成更简单的任务，层层分解，直到最后简单得可以用以上指令来完成。

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