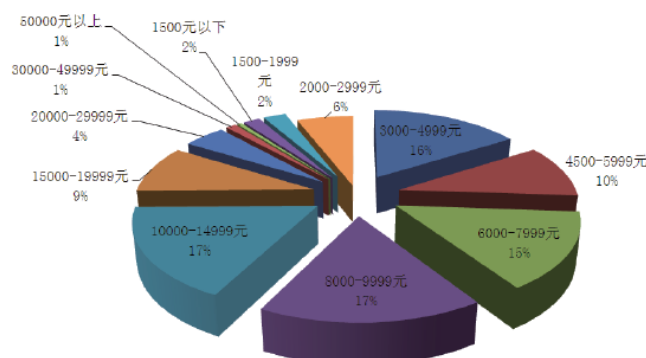


## 为什么选择学习嵌入式?

嵌入式系统无疑是当前最热门最有发展前途的 IT 应用领域之一，同时也是当今 IT 领域仅存的几个金领职位之一。当前的中国 IT 人才面临严重的“后继乏人”，而且这种缺口由于培训缺乏、教育模式等原因造成的，而缺口最大的，就是高级 IT 人才。如果你从事的 IT 培训不专业，面对竞争越来越激烈的职场，基本找不到工作。据专家预测，嵌入式每年人才缺口在 30 万左右。

## 嵌入式行业平均薪资分布



## 嵌入式职业发展讲解视频

视频中主要讲解什么样的人适合从事嵌入式行业、嵌入式行业从业人员需要具备哪些基本素质、嵌入式行业的特点以及嵌入式行业的现状与发展。

● <a href="#">嵌入式研发方向职业生涯讲解视频（1）</a>	● <a href="#">嵌入式研发方向职业生涯讲解视频（2）</a>	● <a href="#">嵌入式研发方向职业生涯讲解视频（3）</a>
● <a href="#">嵌入式研发方向职业生涯讲解视频（4）</a>	● <a href="#">嵌入式研发方向职业生涯讲解视频（5）</a>	

## ARM+Linux 嵌入式**底层内核驱动方向**学习总体路线图



# 基础学习 I ---Linux 入门

目前嵌入式主要开发环境有 Linux、Wince 等；Linux 因其开源、开发操作便利而被广泛采用。而 Linux 操作系统也只是一个简单的操作系统，简单的使用对于嵌入式开发人员来说价值并不很高，真正有价值的是掌握 Linux 的基本服务和 Linux 的设计理念、思想，这对于嵌入式开发人员的长期发展是很极其重要的。Linux 系统有很多发行版，RedHat、Ubuntu、Fedora 等。作为嵌入式开发人员，我们没有必要把精力放到使用哪个 Linux 发行版上，而是尽快把 Linux 系统尽快安装好。如果打算坚持长期学习，那么建议您把自己的电脑做成双系统，而不要在虚拟机上安装。

Ubuntu 系统下载地址：

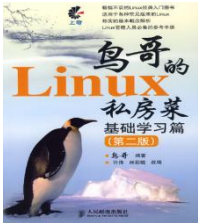
<http://www.ubuntu.org.cn/desktop/get-ubuntu/download/>

Linux



Linus Torvalds  
Linux 核心的创作者

## A) 经典书籍推荐：



《鸟哥的 Linux 私房菜-基础学习篇》

## B) 经典学习视频推荐：

<a href="#">1、Linux 学习建议</a>	<a href="#">2、UNIX like 系统的特点</a>	<a href="#">3、UNIX 原则</a>
<a href="#">4、系统结构与终端控制台</a>	<a href="#">5、RHEL5 Linux 安装</a>	<a href="#">6、Linux 下寻求帮助</a>
<a href="#">7、文件相关命令初步</a>	<a href="#">8、系统命令初步</a>	<a href="#">9、用户及组管理初步</a>
<a href="#">10、文件属性</a>	<a href="#">11、文件夹属性及 umask</a>	<a href="#">12、Linux 编辑器及 VIM 初步</a>
<a href="#">13、Linux 查找及压缩</a>	<a href="#">14、文本处理及正则表达式</a>	<a href="#">15、linux 下 X 图形显示体系</a>
<a href="#">16、进程控制</a>	<a href="#">17、进程优先级前台后台</a>	<a href="#">18、变量快捷键</a>
<a href="#">19、bash 运算符及启动脚本</a>	<a href="#">20、shell 脚本及交互 1</a>	<a href="#">21、shell 脚本及交互 2</a>
<a href="#">22、shell 脚本流程控制</a>	<a href="#">23、shell 脚本 sed&amp;awk</a>	<a href="#">24、Linux 安装软件</a>

## 基础学习 II --- C 语言

C 语言是嵌入式开发必备的基础知识。在 Linux 下从事 C 语言的开发，你会觉得更为顺畅、更为自然，因为 C 语言是因 unix 的出现而诞生的，Linux 内核几乎完全是由 C 语言编写完成的。学习 C 语言，如果不会用指针，那么就



C语言和Unix的发明者、图灵奖得主，  
最具传奇性的程序员Ken Thompson

称不上会 C 语言。做嵌入式开发指针更显得尤为重要，所以做嵌入式开发除了掌握位操作、



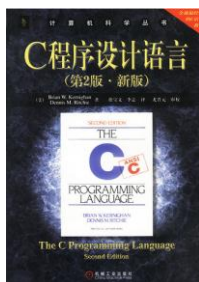
限定词等，对指针的掌握是不可或缺的。而且要掌握多级指针、函数指针等等。涉及到指针，那么就会讲到内存分配。在大学中，学习 C 语言一般的学习很少讲到内存分配，但是如果期望从事嵌入式开发，那么就必须懂得 C 语言是怎么做内存分配管理的。指针之外，还要学习模块化编译处理、指针与数组、gcc、Makefile、GDB、递归、结构体、宏定义使用等。C 语言是整栋嵌入式大厦的基础，所以在学习嵌入式时，必须把 C 语言重视起来，多敲代码，多练。一名优秀的程序员必定是通过“体力劳动”再转向“脑力劳动的”，这也是为什么我们要有“写不出代码=0”思想的原因所在。

C 语言是整栋嵌入式大厦的基础，所以在学习嵌入式时，必须把 C 语言重视起来，多敲代码，多练。一名优秀的程序员必定是通过“体力劳动”再转向“脑力劳动的”，这也是为什么我们要有“写不出代码=0”思想的原因所在。

如果说 C 语言相当于文字，那么数据结构就相当于在造句、写文章。代码质量有一部分取决于对数据结构的掌握程度。在数据结构部分我们要把链表、树、排序作为学习重点。而且我们也可以查看一些比较常见的函数（如 strcpy、strcat、printf 等）在内核中是如何实现的，以及编写代码模拟堆栈，这不仅有利于编写代码质量的提高，而且还可以初步了解 Linux 内核精髓，为今后工作打下坚实的基础。

C++、QT 在嵌入式开发领域，是用来开发图形界面的，希望从事上层应用开发及图形界面，那么在 C++、QT 方面需要比较深入的学习。

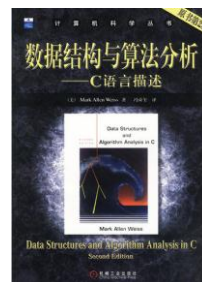
### A) 经典书籍推荐：



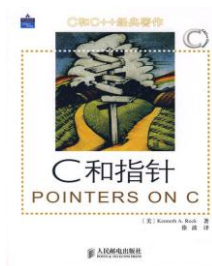
《C 程序设计语言》



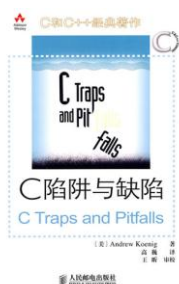
《C 语言核心技术》



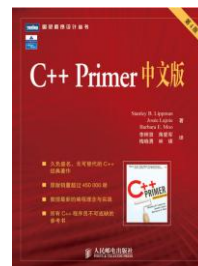
《数据结构与算法分析—C 语言描述》



《C 和指针》



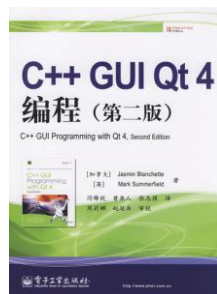
《C 陷阱与缺陷》



《C++ Primer》



《GNU Make》



《C++ GUI Qt4 编程》

## 8) 经典学习视频推荐：

### 1) C 语言学习视频入门篇：

<a href="#">C 语言入门-1、类型运算符表达式</a>	<a href="#">C 语言入门 2、控制流</a>	<a href="#">C 语言入门 3、函数与程序结构</a>
<a href="#">C 语言入门 4、指针与数组</a>	<a href="#">C 语言入门 5、结构体共用体位字段</a>	<a href="#">C 语言入门 6、C 指针</a>

### 2) C 语言学习视频提高篇：

<a href="#">01、C 语言概念、发展历史、特点</a>	<a href="#">2 第一个 helloworld 程序、Linux 下执行</a>	<a href="#">03、C 语言第 2 个程序第 3 个程序</a>
<a href="#">04、C 程序开发过程</a>	<a href="#">05、C 语言开发常见错误分析</a>	<a href="#">06、C 语言开发语言和开发工具的关系</a>
<a href="#">07、C 语言基本数据类型</a>	<a href="#">08、常量和变量</a>	<a href="#">09、const 类型修饰符概念</a>
<a href="#">10、const 类型修饰符实例</a>	<a href="#">11、volatile 类型修饰符</a>	<a href="#">12、extern,static 存储类型说明符</a>
<a href="#">13、register 存储类型说明符</a>	<a href="#">14、auto 存储类型说明符</a>	<a href="#">15、操作符-算术操作符</a>
<a href="#">16、操作符-sizeof 操作符</a>	<a href="#">17、操作符-关系逻辑操作符</a>	<a href="#">18、操作符-位运算操作符</a>
<a href="#">19、表达式</a>	<a href="#">20、if 条件判断结构</a>	<a href="#">21、switch 条件判断结构</a>
<a href="#">22、for 循环结构</a>	<a href="#">23、while 循环结构</a>	<a href="#">24、跳转语句</a>
<a href="#">25、模块化设计方法</a>	<a href="#">26、函数的声明和定义</a>	<a href="#">27、函数的调用</a>
<a href="#">28、递归函数</a>	<a href="#">29、函数返回值</a>	<a href="#">30、数组</a>
<a href="#">31、字符串</a>	<a href="#">32、指针</a>	<a href="#">33、函数指针</a>
<a href="#">34、动态类型分配</a>	<a href="#">35、指针常见错误</a>	<a href="#">36、指针与数组</a>

<a href="#">37、编译预处理</a>	<a href="#">38、结构体基础</a>	<a href="#">39、链表</a>
<a href="#">40、共用体</a>	<a href="#">41、位段</a>	<a href="#">42、枚举</a>
<a href="#">43、文件</a>	<a href="#">44、C 编程总结</a>	

### 3) C++学习视频:

● <a href="#">如何编译 C++ 程序</a>	● <a href="#">2、如何调试 C++ 程序</a>	● <a href="#">3、C++ 预备知识</a>
● <a href="#">4、C++ 中的 C</a>	● <a href="#">5、类和对象</a>	● <a href="#">6、对象的构造和销毁</a>
● <a href="#">7、函数重载和默认参数</a>	● <a href="#">8、内联函数和常量</a>	● <a href="#">9、名字空间</a>
● <a href="#">10、拷贝构造函数 1</a>	● <a href="#">11、拷贝构造函数 2</a>	● <a href="#">12、运算符重载 1</a>
● <a href="#">13、运算符重载 2</a>	● <a href="#">14、动态对象的创建</a>	● <a href="#">15、继承和组合 1</a>
● <a href="#">16、继承和组合 2</a>	● <a href="#">17、多态 1</a>	● <a href="#">18、多态 2</a>
● <a href="#">19、模板 1</a>	● <a href="#">20、模板 2</a>	● <a href="#">21、异常 1</a>
● <a href="#">22、stl</a>	● <a href="#">23、stl-containe</a>	● <a href="#">24、stl-string</a>
● <a href="#">25、makefile</a>	● <a href="#">26、Qt</a>	● <a href="#">27、qtE</a>
● <a href="#">28、qt-extend</a>		

### C) 经典学习案例:

1、学员成绩管理系统 2、计算器 3、电子词典 4、文件压缩器

### 基础进阶---Unix 环境高级编程

Linux 的思想源于 Unix。Linux 继承了 Unix 的优点，但是没有用 UNIX 的代码，所有程序都是重新编写的，而且 Linux 与 Unix 的源码级兼容。Linux 是类 Unix 操作系统。Linux 不仅符合 POSIX 标准，而且还包括其它 UNIX 标准的多种特性，例如，UNIX 的 System V 接口文档 (System V Interface Document, SVID) 和伯克利软件发布 (Berkeley Software Distribution, BSD) 版本。Linux 采用了折衷的策略，包含了 UNIX 几个典型特性当中最实用的一些功能：

Linux 采用了 SVR4 的进程间通信 (IPC) 机制：共享内存、消息队列、信号。

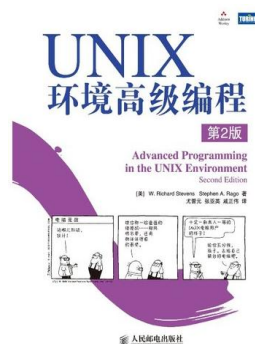
Linux 支持 BSD Socket 网络编程接口。

许多 Linux 发行版采用 SysV init 机制，支持运行级别。

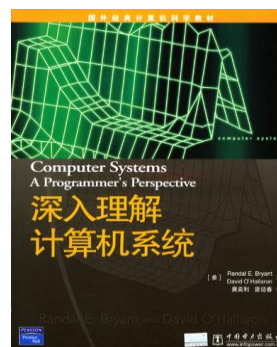
此阶段的学习是从嵌入式上层应用开发以及底层开发人员的必修课程。进程、线

程、信号、文件锁、socket 是这部分内容的重点。要把这些内容学的透彻非要下些苦功夫不可。而且函数是系统提供给我们的，难免要对计算机系统深入理解一番。TCP/IP 协议也是要涉及到的。有问题建议用 Google 搜索及登陆 chinaunix 论坛 (www.chinaunix.net)。通过这个阶段的学习，我们就可以初步实现一些应用程序的开发了，如做自己的 shell 命令解析器、简单的聊天软件、视频多播等。

## A) 经典书籍推荐：



《UNIX 环境高级编程》



《深入理解计算机系统》



《Unix 网络编程》



《TCP/IP 详解》

## B) 经典学习视频推荐：

<a href="#">UNIX 网络开发-1、ARP 攻击原理</a>	<a href="#">UNIX 网络开发-2、ARP 攻击实例分析</a>	<a href="#">UNIX 网络开发-3、ARP 攻击程序代码分析和 socket 构造</a>
--------------------------------------	--	---

## C) 经典学习案例：

1、串口监听程序 2、图片浏览器 3、MP3 播放器 4、聊天程序 5、ftp 程序  
6、音频广播程序 7、ping 的实现 8、DDos 攻击程序 9、ARP 攻击程序



## 嵌入式入门---ARM 体系结构

从这部分开始,我们将真正进入嵌入式开发学习阶段。首先我们要知道采用哪种处理器。目前通用嵌入式处理器有 ARM、MIPS、PowerPC、X86 等。

从市场产品占有率上看,ARM 处理器远远领先于同类其它处理器,并逐步掠夺传统 51 单片机和英特尔市场份额。据招聘网站统计,目前用人单位在技术水平上要求开发人员掌握 ARM9 及以上平台的开发技术。ARM 公司在发布 ARM11 产品后,更改以往的数字标记更新方式,转为发布 ARM Cortex。但是 Cortex 并不是更高端,而是全新系列的处理器,比如 Cortex M3,它主要应用于机械加工、工控领域,目的是取代以前的 ARM7 处理器。

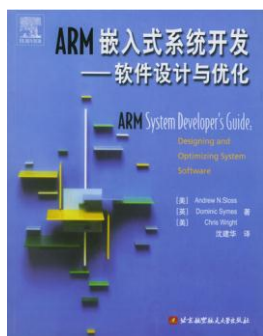
在开始学习前,建议有一块自己的开发板,目前 ARM9 2440 的开发板价格在 400~600 元之间,ARM11 6410 的开发板价格要高一些,最便宜的也要 1200 元以上。

在此,提醒大家尽量不要买 mini 开发板,虽然 mini 开发板的价格要便宜些,但是与正常开发板相比,mini 板功能是不完善的。嵌入式汇编有别于 X86 汇编,所以如果以前没有接触过汇编语言,那么没有必要先去学 X86 汇编再学嵌入式汇编,可以直接学习嵌入式汇编。除了汇编我们还要学习 ARM 指令集。有了开发板,我们要下载相对应处理器的 DataSheet 及开发板电路原理图。ARM 体系结构的学习我们是不能完全照抄别人代码的,因为我们自己手中的开发板及处理器与别人是不同的,除非所用的代码是根据你手中的开发板来编写的。所以,我们要先去看书、看视频资料理解原理及工作方式等,然后根据自己开发板的电路原理图 and DataSheet 来编写代码,达到理解外围设备工作原理和操作硬件的目的。



位于英国剑桥的ARM公司总部

### A) 经典书籍推荐:



《ARM 嵌入式系统开发——软件设计与优化》



《嵌入式Linux应用开发完全手册》

### B) 经典学习视频推荐:

#### 1) 嵌入式扫盲篇

● <a href="#">嵌入式漫谈 1</a>	● <a href="#">嵌入式漫谈 2</a>	● <a href="#">嵌入式漫谈 3</a>
● <a href="#">十问嵌入式 1-什么是嵌入式, 创业</a>	● <a href="#">十问嵌入式 2-嵌入式实例 iphone</a>	● <a href="#">十问嵌入式 3-嵌入式前景</a>
● <a href="#">十问嵌入式 4-嵌入式学习</a>	● <a href="#">十问嵌入式 5-ARM 概述</a>	● <a href="#">十问嵌入式 6-ARM 概述开发板</a>

● <a href="#">十问嵌入式 7-ARM 与单片机</a>	● <a href="#">十问嵌入式 8-汇编语言的作用</a>	● <a href="#">十问嵌入式 9-操作系统的选择</a>
● <a href="#">十问嵌入式 10-编程语言选择</a>		

## 2) 嵌入式开发概述

● <a href="#">嵌入式开发入门经典教程-1</a>	● <a href="#">嵌入式开发入门经典教程-2</a>	● <a href="#">嵌入式开发入门经典教程-3</a>
● <a href="#">嵌入式开发入门经典教程-4</a>	● <a href="#">嵌入式开发入门经典教程-5</a>	● <a href="#">嵌入式开发入门经典教程-6</a>
● <a href="#">嵌入式开发入门经典教程-5</a>	● <a href="#">嵌入式开发入门经典教程-6</a>	● <a href="#">嵌入式开发入门经典教程-7</a>
● <a href="#">嵌入式开发入门经典教程-8</a>	● <a href="#">嵌入式开发入门经典教程-9</a>	● <a href="#">嵌入式开发入门经典教程-10</a>
● <a href="#">嵌入式开发入门经典教程-11</a>	● <a href="#">嵌入式开发入门经典教程-12</a>	● <a href="#">嵌入式开发入门经典教程-13</a>

## 3) ARM 体系结构与 Bootloader 开发

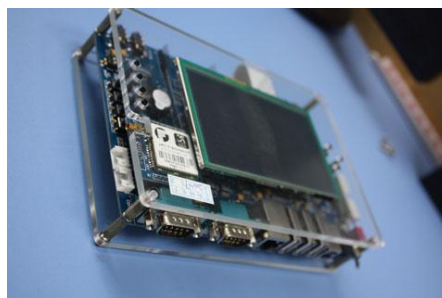
● <a href="#">1、ARM 基础概念介绍与开发板使用教程</a>	● <a href="#">2、编译器高级应用与裸机代码编写教程</a>	● <a href="#">3、ARM 指令与 ARM 汇编入门</a>
● <a href="#">4、ARM 汇编高级教程与 APCS 规范详解</a>	● <a href="#">5、ARM 体系结构详解与异常处理机制</a>	● <a href="#">6、ARM 中断机制详解与 Watchdog 驱动开发实例</a>
● <a href="#">7-1、ARM 外部中断机制详解与 LED 驱动开发实例</a>	● <a href="#">7-2、ARM 嵌入式开发之 ARM 外部中断机制详解与实例</a>	● <a href="#">8、AMBA 总线结构与 ARM 时钟配置</a>
● <a href="#">9、时序图详解与 NAND FLASH 驱动开发实例</a>	● <a href="#">10-1、内存控制器与 SDRAM 原理详解(上)</a>	● <a href="#">10-2、内存控制器与 SDRAM 原理详解(下)</a>
● <a href="#">11、UART 串口详解与通信开发实例</a>	● <a href="#">12、ARM 协处理器 CP15 配置原理</a>	● <a href="#">13、从零开始写 Bootloader 开发过程详解</a>

## C) 经典学习案例:

- |              |                  |               |
|--------------|------------------|---------------|
| 1. ARM 模拟器开发 | 2. Watchdog 控制程序 | 3. RTC 时钟控制程序 |
| 4. Uart 控制程序 | 5. LCD 控制程序      | 6. 触摸屏控制程序    |
| 7. DMA 控制程序  | 8. 音频控制程序        | 9. 网卡控制程序     |
| 10. 文件系统制作   | 11. ARM-Linux 移植 |               |

## 嵌入式提升---Linux 内核、驱动

众所周知嵌入式开发包含至少两个级别，一个是嵌入式内核驱动级别，另外一个为嵌入式应用层开发，而薪酬高的、最具价值的无疑是嵌入式内核驱动级别的开发者。真正的嵌入式高手或者企业中的核心开发人员，一定是嵌入式底层的

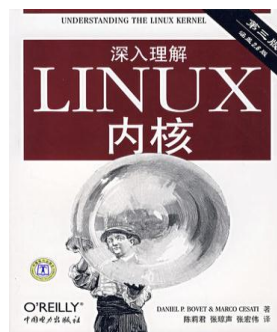




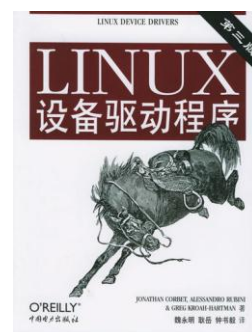
内核驱动开发工程师，这些工程师成长比较缓慢，造成对内核驱动人才的大量需求，所以这些人才是真正企业需要的香饽饽。掌握驱动开发会让自己的核心能力有极大的飞跃，真正实现嵌入式最有价值的就业。

由于 Linux 内核更新速度很快，书籍和视频不可能及时更新，所以我们再看书和视频后，要自己动手下载内核源码包，去看内核中的代码，在这里可以充分检验 C 语言、数据结构等基础知识的掌握程度。而驱动的编写还是要借助外设硬件电路原理图和芯片手册。内核和驱动的知识是让我们根据不同的硬件，编写对应的驱动、合理剪裁内核、制作文件系统，并移植到硬件开发板上。如果手里有 ARM11 6410 及以上的开发板，可以试试在裸板上编写驱动并移植 Android 系统到开发板上吧。

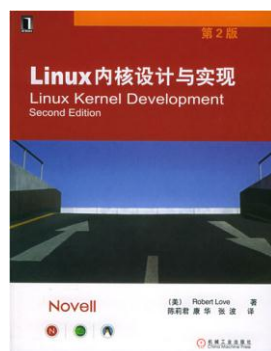
## A) 经典书籍推荐：



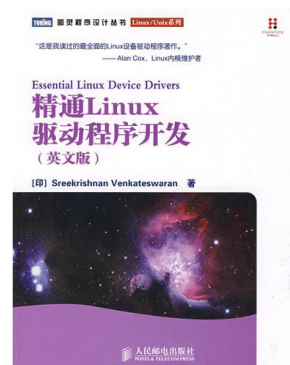
《深入理解 Linux 内核》



《Linux 设备驱动程序》



《Linux 内核设计与实现》



《精通 Linux 驱动程序开发》

## B) 经典学习案例：

1. GPIO 驱动
2. RTC 时钟驱动
3. WatchDog 驱动
4. Udev 移植
5. 触摸屏输入设备驱动
6. LCD 显示设备驱动
7. DMA 驱动
8. 音频设备驱动
9. Nand Flash 块设备驱动
10. 网络设备驱动