

学习嵌入式Linux系统的笔记和体会（1）PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/475/2021\\_2022\\_\\_E5\\_AD\\_A6\\_E4\\_B9\\_A0\\_E5\\_B5\\_8C\\_E5\\_c67\\_475915.htm](https://www.100test.com/kao_ti2020/475/2021_2022__E5_AD_A6_E4_B9_A0_E5_B5_8C_E5_c67_475915.htm) 一个典型的桌面Linux系统包括3个主要的软件层---linux内核、C库和应用程序代码。内核是唯一可以完全控制硬件的层，内核驱动程序代表应用程序与硬件之间进行会话。内核之上是C库，负责把POSIX API转换为内核可以识别的形式，然后调用内核，从应用程序向内核传递参数。应用程序依靠驱动内核来完成特定的任务。在设计嵌入式应用的时候，可以不按照这种层次，应用程序越过C库直接和内核会话，或者把应用和内核捆绑在一起，甚至可以把应用写为内核的一个线程，在内核中运行，虽然这样在移植上带来了困难，但考虑嵌入式系统对尺寸要求小的特点，是完全可行的。不过我们使用三层软件结构的模式来学习嵌入式linux将会是我们认识更清晰，简单可行并使应用具有弹性。快速入门最简单的建立嵌入式Linux应用的方法就是从我们使用的桌面Linux入手，安装一个喜爱的版本，把我们的某个应用作为初始化的一部分，框架就算完成了。当然，嵌入式linux应用远比我们的桌面版本功能简单专一，它也许就是一个用于足彩的终端机，或是一个数码音频播放器，这些系统除了使用嵌入式CPU外，仅仅再需要一个串口，网口等少量的输入输出接口就可以完成它们特定的应用了。在软件上，它可以按照三层的概念由内核装载器，定制的内核和较少的为特定任务设计的静态连接的应用程序组成。之所以使用静态连接的应用程序，是因为少量的静态连接程序所要的存储空间，比同样数量的动态连接

的程序所占的空间小，这个平衡点需要我们在实际开发中去获取。也许你正在设计的是个PDA，它的应用程序较多，那么你很可能就要使用动态连接程序来减少存储空间。在你的/bin或者/sbin目录下，用ll命令看看bash,ifconfig,vi...，也许只用几十K，当你运行ll /bin/bash时，你会看到它们都和好几个库文件相连。好了，这样看来，我们得把PC想像成一个嵌入式硬件平台，再重新制作一个特定功能的嵌入式linux。

基础知识 再进行实际操作之前，先来搞清楚几个基础知识。

内核装载机Loader，它的作用是把内核从外部存储器，移动到内存中。它只作这个事情，一旦完成了调入内核的工作，Loader就跳转到内核位置开始执行。不同架构有不同的Loader，在x86结构的PC上，通常使用的loader有LILO,GRUB,syslinux,syslinux在嵌入式linux中也同样工作。其他非x86架构的应用中，你必须使用专门的loader，或者自己编写loader来装入内核。也有不使用loader的情况，系统加电以后，内核直接从烧录有映像的Flash上开始执行。内核，一旦内核开始执行，它将通过驱动程序初始化所有硬件，这可以从我们的pc机监视器的输出看出来，每个驱动程序都打印一些有关它的信息。初始化完成后，计算机就准备运行嵌入式应用。也许一个，也许是多个应用程序组成了嵌入式应用，但通常首先调用的是init(通过loader向核心传入init=/program可以定制首先运行的程序)。桌面linux中，init会读取/etc/inittab文件，来决定执行级别和哪些脚本和命令。嵌入式应用中，可以根据实际的情况决定是否使用标准的init执行方式，也许这个init是个静态程序，它能够完成我们的嵌入应用的特定任务，那完全不用考虑inittab了。 100Test 下载

频道开通，各类考试题目直接下载。详细请访问

[www.100test.com](http://www.100test.com)