



10年口碑积累，成功培养50000多名研发工程师，铸就专业品牌形象

华清远见的企业理念是不仅要做好良心教育、做专业教育，更要做受人尊敬的职业教育。

《Android 系统移植和驱动开发》

作者：华清远见

专业始于专注 卓识源于远见

第 2 章 Android 系统与嵌入式开发

本章目标

嵌入式开发是当前最热门的领域之一，它无处不在，受到了社会各方面的广泛关注，越来越多的人开始学习嵌入式系统开发，Android 作为当下最热门的手机操作系统，与传统的嵌入式开发既有联系也有区别，本章将向读者介绍嵌入式开发的基本知识，以及它和 Android 系统的联系和区别，主要内容如下：

- 嵌入式系统开发概述。
- 嵌入式系统的概述。
- 嵌入式系统的组成。
- Android 系统的概述。
- Android 系统架构。
- Android 与传统嵌入式 Linux 系统。

专业始于专注 卓识源于远见

2.1 Android 与传统嵌入式 Linux 系统

众所周知，Android 操作系统是基于 Linux 内核的，Google 使用 Linux 内核构建了一个可移植的和健壮的手机平台，而并没有使用传统 Linux 的任何其他东西，基于这个特点，现在从事 Android 开发的人员一共分两大类，一类是从事 Android 应用开发人员，他们使用的开发语言主要是 Java；另一类是 Android 底层开发人员，他们使用的语言主要是 C/C++ 语言。

Android 系统不是完全依赖于 Linux 内核的，这和传统的 Linux 系统还是不一样的，但是从系统移植和驱动开发者的角度来讲，Android 底层的开发移植和传统嵌入式 Linux 系统开发关系非常密切，Android 系统的驱动与 Linux 的驱动在开发上几乎保持了完全一致，另外，Android 底层开发和移植的环境也与嵌入式 Linux 的环境保持了基本一致。

2.2 嵌入式系统概述

2.2.1 嵌入式系统简介

在我们这个时代，嵌入式系统早就广泛应用在科学研究、工程设计、军事技术、各类产业、商业文化艺术、娱乐业，以及人们的日常生活等方方面面，表 2.1 列举了嵌入式系统应用的部分领域。

表 2.1 嵌入式系统应用领域举例

| 领域 | 应用 |
|------|--------------------------|
| 消费电子 | 信息家电 智能玩具 通信设备 移动存储 视频监控 |
| 工业控制 | 工控设备 智能仪表 汽车电子 电子农业 |
| 网络 | 网络设备 电子商务 无线传感器 |
| 医务医疗 | 医疗电子 |
| 军事国防 | 军事电子 |
| 航空航天 | 各类飞行设备、卫星等 |

随着数字信息技术和网络技术的飞速发展，计算机、通信、消费电子的一体化趋势日益明显，这必将培育出一个庞大的嵌入式应用市场。嵌入式系统技术也成了当前关注、学习研究的热点。大家可能会问，究竟什么是嵌入式系统呢？嵌入式系统本身是一个相对模糊的定义，不同的组织对其定义也略有不同，但大意是相同的，我们来看一下嵌入式系统的相关定义。

按照 IEEE 的定义，嵌入式系统是用来控制、监控或辅助操作机器、装置、工厂等大规模系统的设备（devices used to control, monitor, or assist the operation of equipment, machinery or plants）。这主要是从嵌入式系统的用途方面来进行定义的。

更具一般性，也是在多数书籍资料中使用的关于嵌入式系统的定义：嵌入式系统是指以应用为中心，以计算机技术为基础，软件硬件可裁剪，适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

根据以上嵌入式系统的定义，我们可以看出，嵌入式系统是由硬件和软件相结合组成的具有特定功能、用于特定场合的独立系统。其硬件主要由嵌入式微处理器、外围硬件设备组成；其软件主要由底层系统软件 and 用户应用软件组成。

2.2.2 嵌入式系统的特点

1. 专用、软硬件可裁剪、可配置

从嵌入式系统的定义可以看出，嵌入式系统是面向应用的，与通用系统最大的区别在于嵌入式系统功能专一。根据这个特性，嵌入式系统的软、硬件可以根据需要进行精心设计、量体裁衣、去除冗余，以实现低成本、高性能。也正因如此，嵌入式系统采用的微处理器和外围设备种类繁多，系统不具有通用性。

2. 低功耗、高可靠性、高稳定性

嵌入式系统大多用在特定场合，要么是环境条件恶劣，要么要求其长时间连续运转，因此嵌入式系统应具有高可靠性、高稳定性、低功耗等特点。

3. 软件代码短小精悍

由于成本和应用场合的特殊性，通常嵌入式系统的硬件资源（如内存等）都比较少，因此对嵌入式系统设计也提出了较高的要求。嵌入式系统的软件设计要求高质量，要在有限资源上实现高可靠性、高性能的系统。虽然随着硬件技术的发展和成本的降低，在高端嵌入式产品上也开始采用嵌入式操作系统，但其和 PC 资源比起来还是少得可怜，所以，嵌入式系统的软件代码依然要在保证性能的情况下，占用尽量少的资源，保证产品的高性价比，使其具有更强的竞争力。

4. 代码可固化

为了提高执行速度和系统可靠性，嵌入式系统中的软件一般都固化在存储器芯片或单片机本身中，而不是存储于磁盘中。

5. 实时性

很多采用嵌入式系统的应用具有实时性要求，所以大多数嵌入式系统采用实时性系统。但需要注意的是，嵌入式系统不等于实时系统。

6. 弱交互性

嵌入式系统不仅功能强大而且要求使用灵活方便，一般不需要键盘鼠标等。人机交互以简单方便为主。

7. 嵌入式系统软件开发通常需要专门的开发工具和开发环境

嵌入式软件开发有别于桌面软件系统开发的一个显著特点是，它一般需要一个交叉编译和调试环境，即编辑和编译软件在主机上进行（如在 PC 的 Windows 操作系统下），编译好的软件需要下载到目标机上运行（如在一个 PC 目标机上的 VxWorks 操作系统下），主机和目标机建立起通信连接，并传输调试命令和数据。由于主机和目标机往往运行着不同的操作系统，而且处理器的体系结构也彼此不同，这就提高了嵌入式开发的复杂性。

8. 要求开发、设计人员有较高的技能

嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合后的产物。这一点就决定了它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统，从事嵌入式系统开发的人才也必须是复合型人才。

2.2.3 嵌入式系统的发展

1. 在过去的三十多年里，嵌入式系统主要经历了四个阶段

第一阶段是以单芯片为核心的可编程控制器形式的系统。这类系统大部分应用于一些专业性强的工业控制系统中，一般没有操作系统的支持，软件通过汇编语言编写。这一阶段系统的主要特点是：系统结构和功能相对单一，处理效率较低，存储容量较小，几乎没有用户接口。由于这种嵌入式系统使用简单、价格低，因此当时在国内工业领域应用较为普遍，但是现在已经远不能适应高效的、需要大容量存储的现代工业控制和新兴信息家电等领域的需求。

第二阶段是以嵌入式 CPU 为基础、以简单操作系统为核心的嵌入式系统。其主要特点是：CPU 种类繁多，通用性比较弱；系统开销小，效率高；操作系统达到一定的兼容性和扩展性；应用软件较专业化，用户界面不够友好。

第三阶段是以嵌入式操作系统为标志的嵌入式系统。其主要特点是：嵌入式操作系统能运行于各种不同类型的微处理器上，兼容性好；操作系统内核小、效率高，并且具有高度的模块化和扩展性；具备文件和目录管理、支持多任务、支持网络应用、具备图形窗口和用户界面；具有大量的应用程序接口（API），开发应用程序较简单；嵌入式应用软件丰富。

第四阶段是以 Internet 为标志的嵌入式系统。这是一个正在迅速发展的阶段，目前大多数嵌入式系统还孤立于 Internet 之外，但随着 Internet 的发展及 Internet 技术与信息家电、工业控制技术结合日益密切，嵌入式设备与 Internet 的结合将代表嵌入式系统的未来。

2. 未来嵌入式系统的发展趋势

1) 小型化、智能化、网络化、可视化

随着技术水平的提高和人们生活的需要，嵌入式设备（尤其是消费类产品）正朝着小型化便携式和智能化的方向发展。如果携带笔记本电脑外出办事，肯定希望它轻薄小巧，甚至希望有一种更便携的设备来替代它，目前的上网本、MID（移动互联网设备）、便携投影仪等都是应类似的需求而出现的。对嵌入式而言，可以说是已经进入了嵌入式互联网时代（有线网、无线网、广域网和局域网的组合），嵌入式设备和互联网的紧密结合，更为我们的日常生活带来了极大的方便和无限的想象空间。嵌入式设备功能越来越强大，未来我们的冰箱、洗衣机等家用电器都将实现网上控制；异地通信、协同工作、无人操控场所、安全监控场所等的可视化也已经成为了现实，随着网络运载能力的提升，可视化将得到进一步完善。人工智能、模式识别技术也将在嵌入式系统中得到应用，使得嵌入式系统更具人性化、智能化。

2) 多核技术的应用

人们需要处理的信息越来越多，这就要求嵌入式设备运算能力更强，因此需要设计出更强大的嵌入式处理器，多核技术处理器在嵌入式中的应用将更为普遍。

3) 低功耗（节能）、绿色环保

在嵌入式系统的硬件和软件设计中都在追求更低的功耗，以求嵌入式系统能获得更长的可靠工作时间。如手机的通话和待机时间、MP3 听音乐的时间，等等。同时，绿色环保型嵌入式产品将更受人们青睐，在嵌入式系统设计中也会更多的考虑，如辐射和静电等问题。

4) 云计算、可重构、虚拟化等技术被进一步应用到嵌入式系统中

简单来讲，云计算是将计算分布在大量的分布式计算机上，这样只需要一个终端，就可以通过网络服务来实现需要的计算任务，甚至是超级计算任务。云计算（Cloud Computing）是分布式处理（Distributed Computing）、并行处理（Parallel Computing）和网格计算（Grid Computing）的发展，或者说是这些计算机科学概念的商业实现。在未来几年内，云计算将得到进一步发展与应用。

可重构性是指在一个系统中，其硬件模块或（和）软件模块均能根据变化的数据流或控制流对系统结构和算法进行重新配置（或重新设置）。可重构系统最突出的优点就是能够根据不同的应用需求，改变自身的体系结构，以便与具体的应用需求相匹配。

虚拟化是指计算机软件在一个虚拟的平台上而不是真实的硬件上运行。虚拟化技术可以简化软件的重新配置过程，易于实现软件标准化。其中，CPU 的虚拟化可以单 CPU 模拟多 CPU 并行运行，允许一个平台同时运行多个操作系统，并且都可以在相互独立的空间内运行而互不影响，从而提高工作效率和安全性，虚拟化技术是降低多内核处理器系统开发成本的关键。虚拟化技术是未来几年最值得期待和关注的关键技术之一。

随着各种技术的成熟与在嵌入式系统中的应用，将不断为嵌入式系统增添新的魅力和发展空间。

5) 嵌入式软件开发平台化、标准化、系统可升级，代码可复用将更受重视

嵌入式操作系统将进一步走向开放、开源、标准化、组件化。嵌入式软件开发平台化也将是今后的一个趋势，越来越多的嵌入式软、硬件行业标准将出现，最终的目标是使嵌入式软件开发简单化，这也是一个必然规律。同时随着系统复杂度的提高，系统可升级和代码复用技术在嵌入式系统中得到更多的应用。另外，因为嵌入式系统采用的微处理器种类多，不够标准，所以在嵌入式软件开发中将更多地使用跨平台的软件开发语言与工具，目前，Java 语言正在被越来越多地使用到嵌入式软件开发中。

6) 嵌入式系统软件将逐渐 PC 化

需求和网络技术的发展是嵌入式系统发展的一个源动力，随着移动互联网的发展，将进一步促进嵌入式系统软件 PC 化。如前所述，结合跨平台开发语言的广泛应用，那么未来嵌入式软件开发的观念将被逐渐淡化，也就是嵌入式软件开发和非嵌入式软件开发的区别将逐渐减小。

7) 融合趋势

嵌入式系统软硬件融合、产品功能融合、嵌入式设备和互联网的融合趋势加剧。嵌入式系统设计中软硬件结合将更加紧密，软件将是其核心。消费类产品将在运算能力和便携方面进一步融合。传感器网络迅速发展，将极大地促进嵌入式技术和互联网技术的融合。

8) 安全性

随着嵌入式技术和互联网技术的结合发展，嵌入式系统的信息安全问题日益凸显，保证信息安全也成为了嵌入式系统开发的重点和难点。

2.3 嵌入式系统的组成

由前面的介绍我们可以知道，嵌入式系统总体上是由硬件和软件组成的，硬件是基础，软件是核心与灵魂。它们之间的关系如图 2.1 所示。

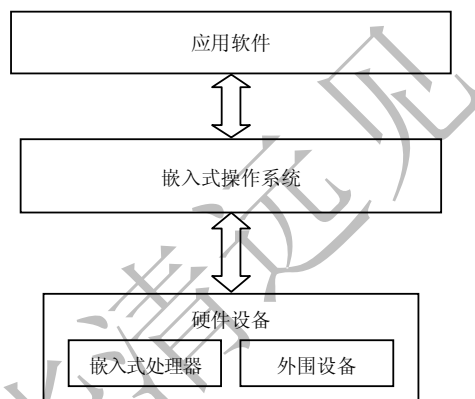


图 2.1 嵌入式系统结构简图

2.3.1 嵌入式系统的硬件组成

嵌入式系统硬件设备包括嵌入式处理器和外围设备。其中，嵌入式处理器（CPU）是嵌入式系统的核心部分，它与通用处理器最大的区别在于，嵌入式处理器大多工作在为特定用户群所专门设计的系统中，它将通用处理器中许多由板卡完成的任务集成到芯片内部，从而有利于嵌入式系统在设计时趋于小型化，同时还具有很高的效率和可靠性。如今，全世界嵌入式处理器已经超过 1000 多种，流行的体系结构有 30 多个系列，其中以 ARM、PowerPC、MC 68000、MIPS 等使用得最为广泛。

外围设备是嵌入式系统中用于完成存储、通信、调试、显示等辅助功能的其他部件。目前常用的嵌入式外围设备按功能可以分为存储设备（如 RAM、SRAM、Flash 等）、通信设备（如 RS-232 接口、SPI 接口、以太网接口等）和显示设备（如显示屏等）3 类。

常见存储器概念辨析：RAM、SRAM、SDRAM、ROM、EPROM、EEPROM、Flash。

存储器可以分为很多种类，其中根据掉电后数据是否丢失可以分为 RAM（随机存取存储器）和 ROM（只读存储器），其中 RAM 的访问速度比较快，但掉电后数据会丢失，而 ROM 掉电后数据不会丢失。人们通常所说的内存即指系统中的 RAM。

RAM 又可分为 SRAM（静态存储器）和 DRAM（动态存储器）。SRAM 是利用双稳态触发器来保存信息的，只要不掉电，信息是不会丢失的。DRAM 是利用 MOS（金属氧化物半导体）电容存储电荷来存储信息的，因此必须通过不停地给电容充电来维持信息，所以 DRAM 的成本、集成度、功耗等明显优于 SRAM。

而通常人们所说的 SDRAM 是 DRAM 的一种，它是同步动态存储器，利用一个单一的系统时钟同步所有的地址数据和控制信号。使用 SDRAM 不但能提高系统表现，还能简化设计、提供高速的数据传输。在嵌入式系统中经常使用。

EPROM、E²PROM 都是 ROM 的一种，分别为可擦除可编程 ROM 和电可擦除 ROM，但使用不是很方便。

Flash 也是一种非易失性存储器（掉电不会丢失），它擦写方便，访问速度快，已大大取代了传统的 EPROM 的地位。由于它具有和 ROM 一样掉电不会丢失的特性，因此很多人称其为 Flash ROM。

2.3.2 嵌入式系统的软件组成

在嵌入式系统不同的应用领域和不同的发展阶段，嵌入式系统软件组成也不完全相同，大致如图 2.2 所示。

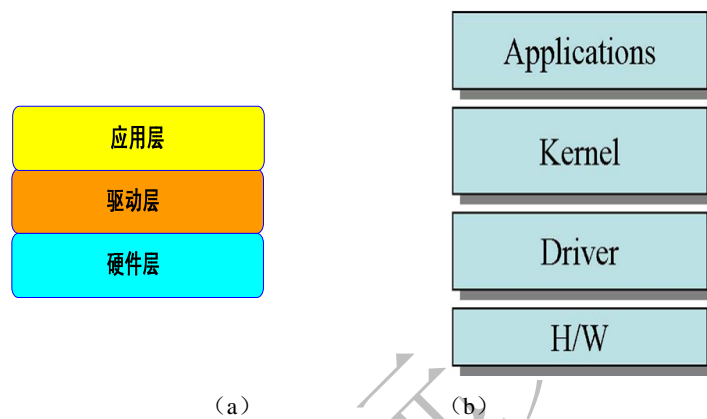


图 2.2 嵌入式系统软件组成图

如图 2.2 (a) 所示，在某些特殊领域，嵌入式系统软件没有使用如通用计算机系统的。嵌入式操作系统从嵌入式发展的第三阶段起开始引入。嵌入式操作系统不仅具有通用操作系统的一般功能，如向上提供对用户的接口（如图形界面、库函数 API 等），向下提供与硬件设备交互的接口（硬件驱动程序等），管理复杂的系统资源，同时，它还在系统实时性、硬件依赖性、软件固化性及应用专用性等方面，具有更加鲜明的特点。

应用软件是针对特定应用领域，基于某一固定的硬件平台，用来达到用户预期目标的计算机软件。由于嵌入式系统自身的特点，决定了嵌入式应用软件不仅要求做到准确性、安全性和稳定性等方面的需要，而且还要尽可能地进行代码优化，以减少对系统资源的消耗，降低硬件成本。

2.4 嵌入式系统开发概述

由嵌入式系统本身的特性所影响，嵌入式系统开发与通用系统的开发有很大的区别。嵌入式系统的开发主要分为系统总体开发、嵌入式硬件开发和嵌入式软件开发三大部分，其总体流程图如图 2.3 所示。

在系统总体开发中，由于嵌入式系统与硬件依赖程序非常紧密，往往某些需求只能通过特定的硬件才能实现，因此需要进行处理器选型，以更好地满足产品的需求。另外，对于有些硬件和软件都可以实现的功能，就需要在成本和性能上做出选择。往往通过硬件实现会增加产品的成品，但能大大提高产品的性能和可靠性。

再次，开发环境的选择对于嵌入式系统的开发也有很大的影响。这里的开发环境包括嵌入式操作系统的选择及开发工具的选择等。2.2 节对各种不同的嵌入式操作系统进行了比较，读者可以以此为依据进行相关的选择。例如，对开发成本和进度限制较大的产品可以选择嵌入式 Linux，对实时性要求非常高的产品可以选择 VxWorks 等。

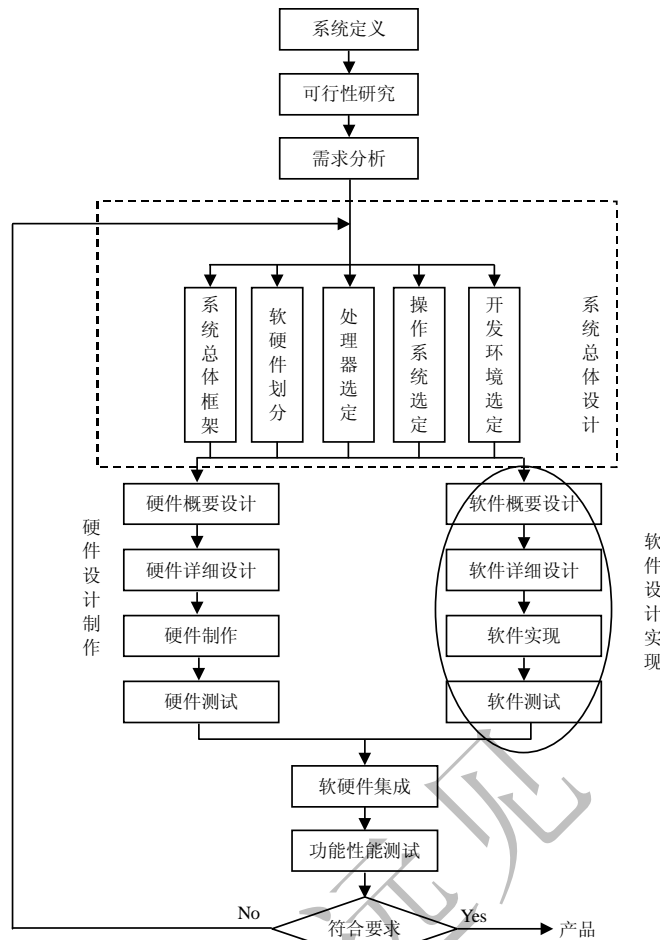


图 2.3 嵌入式系统开发流程图

嵌入式软件开发总体流程为图 2.3 中“软件设计实现”部分所示，它同通用计算机软件开发一样，分为需求分析、软件概要设计、软件详细设计、软件实现和软件测试。其中嵌入式软件需求分析与硬件的需求分析合二为一，故没有分开画出。

由于嵌入式开发的工具非常多，为了更好地帮助读者选择开发工具，下面首先对嵌入式软件开发过程中所使用的工具进行简单归纳。

嵌入式软件的开发工具根据不同的开发过程划分，例如，在需求分析阶段，可以选择 IBM 的 Rational Rose 等软件；而在程序开发阶段可以采用 CodeWarrior（下面要介绍的 ADS 的一个工具）等；在调试阶段所用的 Multi-ICE 等。同时，不同的嵌入式操作系统往往会有配套的开发工具，如 VxWorks 的集成开发环境 Tornado、WinCE 的集成开发环境 WinCE Platform 等。此外，不同的处理器可能还有针对的开发工具，如 ARM 的常用集成开发工具 ADS 等。在这里，大多数软件都有比较高的使用费用，但也可以大大加快产品的开发进度，用户可以根据需求自行选择。如图 2.4 所示为嵌入式开发的不同阶段的常用软件。

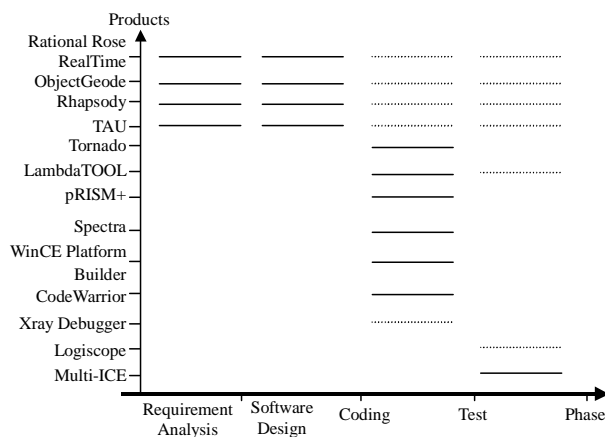


图 2.4 嵌入式开发不同阶段的常用软件

嵌入式系统的软件开发与通常软件开发的区别主要在于软件实现部分，其中又可以分为编译和调试两部分，下面分别对这两部分进行讲解。

1. 交叉编译

嵌入式软件开发所采用的编译为交叉编译。所谓交叉编译就是在一个平台上生成可以在另一个平台上执行的代码。因此，不同的 CPU 需要有相应的编译器，而交叉编译就如同翻译一样，把相同的程序代码翻译成不同的 CPU 对应语言。需要注意的是，编译器本身也是程序，也要在与之对应的某一个 CPU 平台上运行。

一般把进行交叉编译的主机称为宿主机，也就是普通的通用计算机，而把程序实际的运行环境称为目标机，也就是嵌入式系统环境。由于一般通用计算机拥有非常丰富的系统资源、使用方便的集成开发环境和调试工具等，而嵌入式系统的系统资源非常紧缺，没有相关的编译工具，因此，嵌入式系统的开发需要借助宿主机（通用计算机）来编译出目标机的可执行代码。

由于编译的过程包括编译、链接等几个阶段，因此，嵌入式的交叉编译也包括交叉编译、交叉链接等过程，通常 ARM 的交叉编译器为 `arm.elf.gcc`，交叉链接器为 `arm.elf.ld`，交叉编译过程如图 2.5 所示。

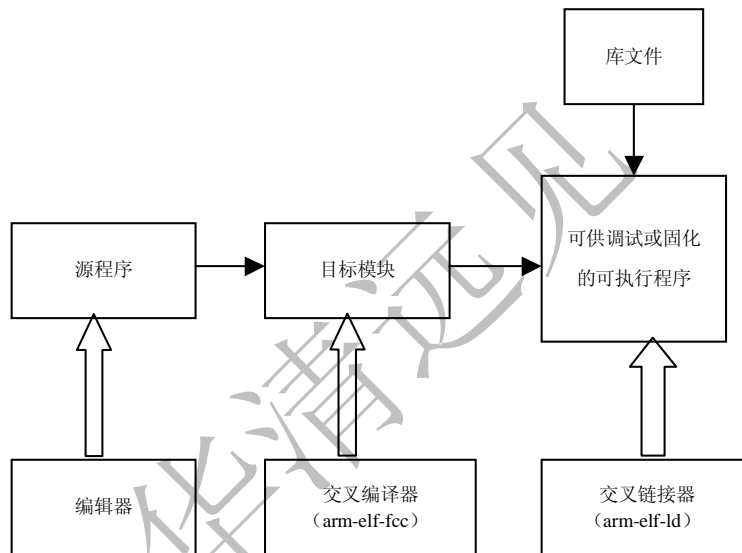


图 2.5 嵌入式交叉编译过程

2. 交叉调试

嵌入式软件经过编译和链接后即进入调试阶段，调试是软件开发过程中必不可少的一环，嵌入式软件开发过程中的交叉调试与通用软件开发过程中的调试方式有很大的差别。在常见软件开发中，调试器与被调试的程序往往运行在同一台计算机上，调试器是一个单独运行着的进程，它通过操作系统提供的调试接口来控制被调试的进程。而在嵌入式软件开发中，调试时采用的是在宿主机和目标机之间进行的交叉调试，调试器仍然运行在宿主机的通用操作系统之上，但被调试的进程却运行在基于特定硬件平台的嵌入式操作系统中，调试器和被调试进程通过串口或者网络进行通信，调试器可以控制、访问被调试进程，读取被调试进程的当前状态，并能够改变被调试进程的运行状态。

嵌入式系统的交叉调试有多种方法，主要可分为软件方式和硬件方式两种。它们一般都具有如下一些典型特点。

- ❑ 调试器和被调试进程运行在不同的机器上，调试器运行在 PC 或者工作站上（宿主机），而被调试的进程则运行在各种专业调试板上（目标机）。
- ❑ 调试器通过某种通信方式（串口、并口、网络、JTAG 等）控制被调试进程。
- ❑ 在目标机上一般会具备某种形式的调试代理，它负责与调试器共同配合完成对目标机上运行着的进程的调试。这种调试代理可能是某些支持调试功能的硬件设备，也可能是某些专门的调试软件（如 GDBServer）。

- 目标机可能是某种形式的系统仿真器，通过在宿主机上运行目标机的仿真软件，整个调试过程可以在一台计算机上运行。此时物理上虽然只有一台计算机，但逻辑上仍然存在着宿主机和目标机的区别。

下面分别就软件调试桩方式和硬件片上调试两种方式进行详细介绍。

1) 软件调试

软件调试方式主要是通过插入调试桩的方式来进行的。调试桩方式进行调试是通过目标操作系统和调试器内分别加入某些功能模块，二者互通信息来进行调试。该方式的典型调试器有 GDB 调试器。

GDB 的交叉调试器分为 GDBServer 和 GDBClient，其中 GDBServer 作为调试桩安装在目标板上，GDBClient 即驻于本地的 GDB 调试器。它们的调试原理图如图 2.6 所示。

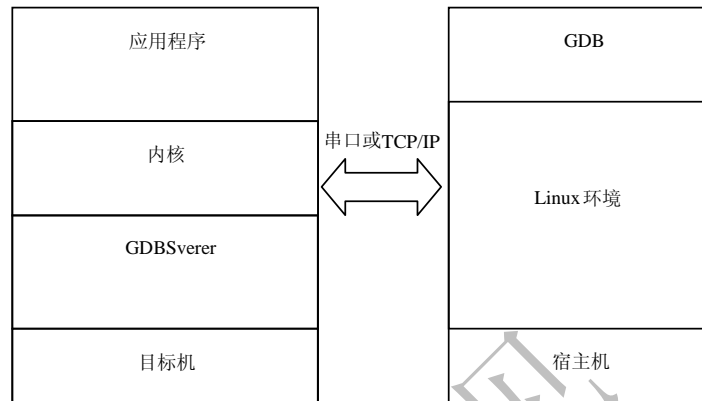


图 2.6 GDB 远程调试原理图

GDB 调试桩的工作流程如下：

- (1) 建立调试器（本地 GDB）与目标操作系统的通信连接，可通过串口、网卡、并口等多种方式。
- (2) 在目标机上开启 GDBServer 进程，并监听对应端口。
- (3) 在宿主机上运行调试器 GDB，这时，GDB 就会自动寻找远端的通信进程，也就是 GDBServer 所在的进程。

(4) 在宿主机的 GDB 通过 GDBServer 请求对目标机上的程序发出控制命令。这时，GDBServer 将请求转化为程序的地址空间或目标平台的某些寄存器的访问，这对于没有虚拟存储器的简单嵌入式操作系统而言，是十分容易的。

(5) GDBServer 把目标操作系统的所有异常处理转向通信模块，并告知宿主机上 GDB 当前异常。宿主机的 GDB 向用户显示被调试程序产生了哪类异常。

这样就完成了调试的整个过程。这个方案的实质是用软件接管目标机的全部异常处理及部分中断处理，并在其中插入调试端口通信模块，与主机的调试器进行交互。但是它只能在目标机系统初始化完毕、调试通信端口初始化完成后才能起作用，因此，一般只能用于调试运行于目标操作系统之上的应用程序，而不宜用来调试目标操作系统的内核代码及启动代码。而且，它必须改变目标操作系统，因此，也就多了一个不用于正式发布的调试版。

2) 硬件调试

相对于软件调试而言，使用硬件调试器可以获得更强大的调试功能和更优秀的调试性能。硬件调试器的基本原理是通过仿真硬件的执行过程，让开发者在调试时可以随时了解到系统的当前执行情况。目前，嵌入式系统开发中最常用到的硬件调试器是 ROMMonitor、ROMEmulator、In.Circuit Emulator 和 In.Circuit Debugger。

- 采用 ROMMonitor 方式进行交叉调试需要在宿主机上运行调试器，在目标机上运行 ROM 监视器（ROMMonitor）和被调试程序，宿主机通过调试器与目标机上的 ROM 监视器遵循远程调试协议建立通信连接。ROM 监视器可以是一段运行在目标机 ROM 上的可执行程序，也可以是一个专门的硬件调试设备，它负责监控目标机上被调试程序的运行情况，能够与宿主机端的调试器一同完成对应用程序的调试。

在使用这种调试方式时，被调试程序首先通过 ROM 监视器下载到目标机，然后在 ROM 监视器的监控下完成调试。

优点：ROM 监视器功能强大，能够完成设置断点、单步执行、查看寄存器、修改内存空间等各项调试功能。

确定：同软件调试一样，使用 ROM 监视器目标机和宿主机必须建立通信连接。

其原理如图 2.7 所示。

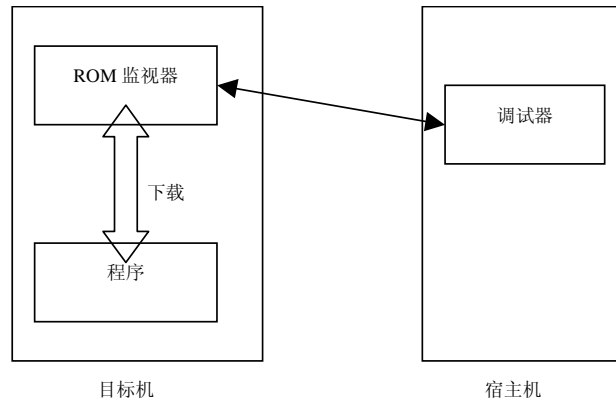


图 2.7 ROMMonitor 调试方式

□ 采用 ROMEmulator 方式进行交叉调试时需要使用 ROM 仿真器，并且它通常被插入到目标机上的 ROM 插槽中，专门用于仿真目标机上的 ROM 芯片。

在使用这种调试方式时，被调试程序首先下载到 ROM 仿真器中，因此等效于下载到目标机的 ROM 芯片上，然后在 ROM 仿真器中完成对目标程序的调试。

优点：避免了每次修改程序后都必须重新烧写到目标机的 ROM 中。

缺点：ROM 仿真器本身比较昂贵，功能相对来讲又比较单一，只适应于某些特定场合。

其原理如图 2.8 所示。

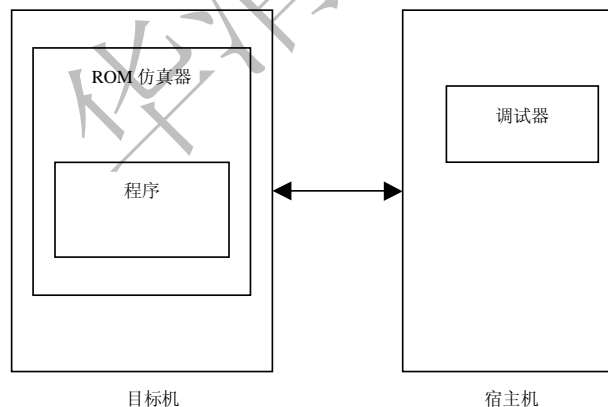


图 2.8 ROMEmulator 调试方式

□ 采用 In.Circuit Emulator (ICE) 方式进行交叉调试时需要使用在线仿真器，它是目前最为有效的嵌入式系统的调试手段。它是仿照目标机上的 CPU 而专门设计的硬件，可以完全仿真处理器芯片的行为。仿真器与目标板可以通过仿真头连接，与宿主机可以通过串口、并口、网线或 USB 口等连接方式。由于仿真器自成体系，所以调试时既可以连接目标板，也可以不连接目标板。在线仿真器提供了非常丰富的调试功能。在使用在线仿真器进行调试的过程中，可以按顺序单步执行，也可以倒退执行，还可以实时查看所有需要的数据，从而给调试过程带来了很大便利。嵌入式系统应用的一个显著特点是与现实世界中的硬件直接相关，并存在各种异变和事先未知的变化，从而给微处理器的指令执行带来各种不确定因素，这种不确定性在目前情况下只有通过在线仿真器才有可能发现。

优点：功能强大，软硬件都可做到完全实时在线调试。

缺点：价格昂贵。

其原理如图 2.9 所示。

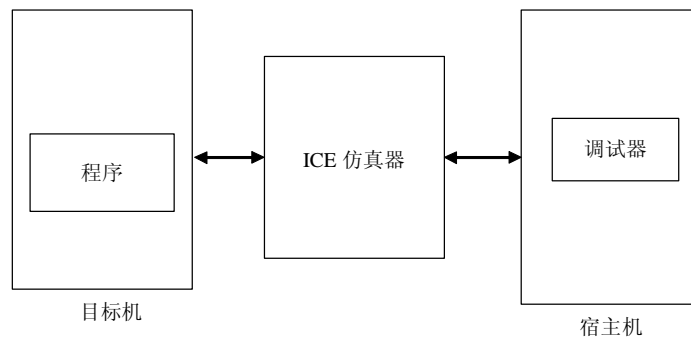


图 2.9 ICE 调试方式

□ 采用 In.Circuit Debugger (ICD) 方式进行交叉调试时需要使用在线调试器。由于 ICE 的价格非常昂贵，并且每种 CPU 都需要一种与之对应的 ICE，使得开发成本非常高。一个比较好的解决办法是让 CPU 直接在其内部实现调试功能，并通过在开发板上引出的调试端口发送调试命令和接收调试信息，完成调试过程。如应用非常广泛的 ARM 处理器的 JTAG 端口技术就是由此而诞生的。

JTAG 是 1985 年制定的检测 PCB 和 IC 芯片的一个标准。1990 年被修改成为 IEEE 的一个标准，即 IEEE1149.1。JTAG 标准所采用的主要技术为边界扫描技术，它的基本思想就是在靠近芯片的输入/输出引脚上增加一个移位寄存器单元。因为这些移位寄存器单元都分布在芯片的边界上（周围），所以被称为边界扫描寄存器（Boundary.Scan Register Cell）。

当芯片处于调试状态时，这些边界扫描寄存器可以将芯片和外围的输入/输出隔离开来。通过这些边界扫描寄存器单元，可以实现对芯片输入/输出信号的观察和控制。对于芯片的输入引脚，可通过与之相连的边界扫描寄存器单元把信号（数据）加载到该引脚中去；对于芯片的输出引脚，可以通过与之相连的边界扫描寄存器单元“捕获”（Capture）该引脚的输出信号。这样，边界扫描寄存器提供了一个便捷的方式用于观测和控制所需要调试的芯片。

现在较为高档的微处理器都带有 JTAG 接口，包括 ARM7、ARM9、StrongARM、DSP 等，通过 JTAG 接口可以方便地对目标系统进行测试，同时，还可以实现 Flash 的编程，是非常受欢迎的。

优点：连接简单，成本低。

缺点：特性受制于芯片厂商。

其原理如图 2.10 所示。

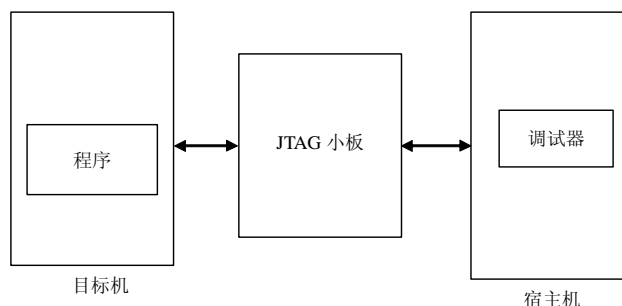


图 2.10 JTAG 调试方式

2.5 Android 系统概述

Android 一词英文本义是指“机器人”，它是由 Google 公司于 2007 年 11 月宣布的基于 Linux 平台的开源手机操作系统，依靠 Google 的强大开发和媒体资源，Android 成为众多手机厂商竞相追逐的对象，逐渐发展成为目前最流行的手机开发平台。如图 2.11 所示为 Android 系统的 Logo。



图 2.11 Android 系统的 Logo

Android 是一个包括操作系统、中间件、用户界面和关键应用程序的移动设备软件堆。换句话说，就是 Android 是基于 Java 并运行在 Linux 内核上的轻量级操作系统，功能全面，包括一系列 Google 公司在其上内置的应用程序，如电话、短信等基本应用功能。如图 2.12 所示为 Android 的模拟器，从中可以大概了解 Android 的运行界面。

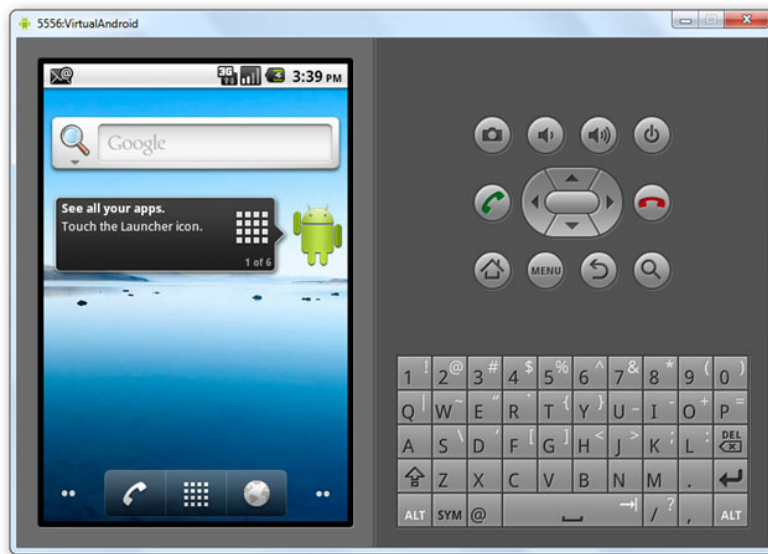


图 2.12 Android 模拟器



问：模拟器和真机有什么区别呢？

答：模拟器不支持呼叫和接听实际来电，但可以通过控制台模拟电话呼叫（呼入和呼出）。

模拟器不支持 USB 连接。

模拟器不支持相机/视频捕捉。

模拟器不支持音频输入（捕捉），但支持输出（重放）。

模拟器不支持扩展耳机。

模拟器不能确定连接状态。

模拟器不能确定电池电量水平和充电状态。

模拟器不能确定 SD 卡的插入/弹出。

模拟器不支持蓝牙。

2.6 Android 系统架构

Android 系统从下至上分为 4 层：Linux 内核、Android 核心库及 Android 运行时环境（Android Runtime）、应用程序框架、应用程序及小部件，如图 2.13 所示。



图 2.13 Android 模拟器

2.6.1 Linux 内核 (Linux Kernel)

Android 的核心系统服务依赖于 Linux 2.6, 例如, 安全、内存管理、进程管理、网络堆栈、驱动模型。Linux Kernel 也作为硬件和软件之间的抽象层, 它隐藏具体硬件细节而为上层提供统一的服务。

除了标准的 Linux 内核以外, Android 系统还增加了内核的驱动程序, 例如, 显示驱动、蓝牙驱动、相机驱动、闪存卡驱动、Binder IPC 驱动、输入设备驱动、USB 驱动、WiFi 驱动、音频系统驱动、电源管理等, 为 Android 系统的运行提供基础性支持。

这样分层的好处就是使用下层提供的服务而为上层提供统一的服务, 屏蔽本层及以下层的差异, 当本层及以下层发生了变化时不会影响到上层。也就是说各层各尽其职, 各层提供固定的 SAP (Service Access Point), 即高内聚、低耦合。

2.6.2 Android 核心库 (Libraries)

Android 包含一个 C/C++库的集合, 以供 Android 系统的各个组件使用。这些功能通过 Android 的应用程序框架 (Application Framework) 暴露给开发者。下面列出一些核心库。

- ❑ 系统 C 库: 由 BSD 继承衍生的标准 C 系统函数库 (libc), 调整为基于嵌入式 Linux 设备的库。
- ❑ 媒体库: 基于 PacketVideo 的 OpenCore。这些库支持播放和录制多种流行的音频和视频格式, 以及多种媒体格式的编码/解码格式, 包括 MPEG4、H.264、MP3、AAC、AMR、JPG、PNG。
- ❑ 界面管理: 显示子系统的管理器, 管理访问显示子系统和无缝组合多个应用程序的二维和三维图形层。
- ❑ LibWebCore: 新式的 Web 浏览器引擎, 驱动 Android 浏览器和可嵌入的 Web 视图。
- ❑ SGL: Skia 图形库, 基本的 2D 图形引擎。
- ❑ 3D libraries: 基于 OpenGL ES APIs 的实现。该库使用硬件 3D 加速或使用高度优化的 3D 软加速。
- ❑ FreeType: 位图 (Bitmap) 和矢量 (Vector) 字体渲染。
- ❑ SQLite: 所有应用程序都可以使用的强大而轻量级的关系数据库引擎。

2.6.3 Android 运行时环境 (Android Runtime)

在 Linux 内核层上还有一个 Android 运行时层，该层包括 Dalvik 虚拟机及 Java 核心库，提供了 Java 编程语言核心库的大多数功能。

Dalvik 虚拟机是 Android 使用的 Java 虚拟机。每一个 Android 应用程序是 Dalvik 虚拟机中的实例，运行在它们自己的进程中。Dalvik 虚拟机设计成在一个设备中可以高效地运行多个虚拟机。Dalvik 虚拟机可执行文件格式是 .dex，dex 格式是专为 Dalvik 设计的一种压缩格式，适合内存和处理器速度有限的系统。

大多数虚拟机包括 JVM 都是基于栈的，而 Dalvik 虚拟机则是基于寄存器的。两种架构各有优劣，一般而言，基于栈的机器需要更多指令，而基于寄存器的机器指令更大。dx 是一套工具，可以将 Java .class 转换成 .dex 格式。一个 .dex 文件通常会有多个 .class。由于 .dex 有时必须进行最佳化，会使文件大小增加 1.4 倍，以 ODEX 结尾。

Dalvik 虚拟机依赖于 Linux 内核提供基本功能，如线程和底层内存管理。

2.6.4 Android 应用程序框架（Application Framework）

位于 Android 程序库和运行时上面的是应用程序框架层。通过提供开放的开发平台，Android 使开发者能够访问核心应用程序所使用的 API 框架，这样使得组件的重用得以简化，任何应用程序都能发布它的功能且任何其他应用程序可以使用这些功能（需要服从框架执行的安全限制）。从而使得开发者可以编制极其丰富和新颖的应用程序，自由地利用设备硬件优势、访问位置信息、运行后台服务、设置闹钟、向状态栏添加通知等。

每个应用程序其实是一组服务和系统，包括如下几项。

- ❑ 视图 (View): 丰富的、可扩展的视图集合，用来构建应用程序。包括列表 (ListView)、网格 (Grid)、文本框 (EditText/TextView)、按钮 (Button) 等，甚至是可嵌入的网页浏览器 (WebView)。
- ❑ 内容提供者 (Content Providers): 使应用程序可以访问其他应用程序（如通讯录）的数据，或共享自己的数据。
- ❑ 资源管理器 (Resource Manager): 提供对于非代码资源的访问，如本地化字符串、图形和布局文件。
- ❑ 通知管理器 (Notification Manager): 使得应用程序能够在状态栏显示自定义的提示信息。
- ❑ 活动管理器 (Activity Manager): 管理应用程序生命周期，并提供常用的导航回退功能。

2.6.5 Android 应用程序和小部件

Android 装配一个核心应用程序集合，连同系统一起发布，这些应用程序包括电子邮件客户端、SMS 程序、日历、地图、浏览器、联系人和其他设置等。所有应用程序都是用 Java 语言编写的，由用户开发的 Android 应用程序和 Android 核心应用程序是同一层次的。

2.7 小结

本章主要介绍了传统嵌入式开发和 Android 系统之间的关系，如果以前是一个嵌入式软件工程师，那么阅读了本章后就会有一种恍然大悟的感觉，从而更快了解 Android 系统。另外，本章在前面的基础上分析了 Android 系统的层次结构，说明了 Android 系统和嵌入式 Linux 系统之间的关系。

2.8 思考题

1. 什么是嵌入式系统？列举几个身边熟悉的嵌入式系统产品。
2. 嵌入式系统由哪几部分组成？

3. 列举出 Android 系统的 4 个层次。
4. 简述嵌入式系统的特点。

联系方式

集团官网: www.hqyj.com 嵌入式学院: www.embedu.org 移动互联网学院: www.3g-edu.org

企业学院: www.farsight.com.cn 物联网学院: www.topsight.cn 研发中心: dev.hqyj.com

集团总部地址: 北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址: 北京市海淀区西三旗悦秀路北京明园大学校区, 电话: 010-82600386/5

上海地址: 上海市徐汇区漕溪路 250 号银海大厦 11 层 B 区, 电话: 021-54485127

深圳地址: 深圳市龙华新区人民北路美丽 AAA 大厦 15 层, 电话: 0755-22193762

成都地址: 成都市武侯区科华北路 99 号科华大厦 6 层, 电话: 028-85405115

南京地址: 南京市白下区汉中路 185 号鸿运大厦 10 层, 电话: 025-86551900

武汉地址: 武汉市工程大学卓刀泉校区科技孵化器大楼 8 层, 电话: 027-87804688

西安地址: 西安市高新区高新一路 12 号创业大厦 D3 楼 5 层, 电话: 029-68785218

广州地址: 广州市天河区中山大道 268 号天河广场 3 层, 电话: 020-28916067

