

# 嵌入式系统原理——交叉编译环境的构建

## 说课稿

教师：刘海波

### 一、教材

#### 1. 对教材的选取与处理

本课内容所选的教材来自两部分：一是嵌入式系统原理这门课程所选教材《嵌入式系统原理与应用——基于 ARM 微处理器和 Linux 操作系统》第 3 章：Linux 系统编程基础的部分内容及第四章：嵌入式交叉编译环境及系统裁切的部分内容；二是作为对课本中交叉编译环境的构建实践操作缺乏的补充，使用

<http://cross-lfs.org/view/clfs-embedded/arm/cross-tools/chapter.html> 开源社区所提供的交叉编译环境构建方案作为学生实践环境的操作指南。

#### 2. 本节内容所处地位

本节课内容安排在对 ARM 处理器及其编程环境的介绍之后，在学生理解了 ARM 架构和 x86 架构的巨大差异之后，才具备了理解“交叉编译”概念内涵的基础；而通过构建交叉编译环境的训练，学生能够掌握在与目标平台（ARM）异构的平台（x86）之上进行软件开发实践的基本工具和原理，从而为整个后续课程：包括启动加载器、内核、根文件系统、Android 系统的交叉编译打下坚实的基础。故本课是整门课程承上启下的重要环节，是重中之重。

#### 3. 对学生知识能力培养的作用

从知识上讲，通过对本课的学习。学生可以从实践角度对软件编译的基本环节有一个清楚的认识，这对于没有经过编译原理课程进行理论地，系统地学习的学生而言，是一个为今后进行的大量软件编译工程实践夯实理论基础的机会；从能力上讲，经过交叉编译环境构建的大量 Linux 命令行环境的实践操作，对学生使用 Linux 开源软件开发环境的能力是一个极大的提高。

## 4. 对将来学习的影响

通过本课的学习，学生对 Linux 软件开发环境下的软件开发操作不再陌生和惧怕，从而为后续课程实践环节中反复使用本课所讲授的知识和技能奠定基础。教学重点与难点的确立

本课在教学大纲中的基本要求有以下几点：

- 1) 掌握 Linux 编译环境环境的构成：Gcc 、Make 和 Autotools；
  - 2) 以“Helloworld”为例，理解 GCC 编译的基本步骤（既基本原理）；
  - 3) 通过对比前述的本机编译，掌握交叉编译的基本原理；
  - 4) 掌握交叉编译环境的构成以及使用开源技术构建一个交叉编译环境的方法。
- 其中，交叉编译的基本原理和构建交叉编译环境的方法，既是本课的教学重点，同是也是难点。

只有理解了交叉编译的原理，学生才不会在后续的实际操作中面对复杂的构建步骤和重复使用但差别细微的一系列命令感到毫无头绪一团乱麻；而只有掌握了交叉编译环境的构建过程，学生才能在今后的工程实践中自如地应对交叉编译过程中可能出现的各种问题。这是本课将此点作为重点的原因。

而对学生而言，由于大部分学生缺乏编程实践，尤其是 Linux 开源环境下的编程实践；再加上学生对 Linux 命令行环境的生疏，这些都让本课该知识点/技能点的学习变得格外的困难。

## 5. 教学的逻辑结构和教学体系

为了克服上述困难，在教学的逻辑结构上，本课从知识点的联系与对比着手，从一个最简单的“Helloworld”的 C 语言源程序编译成可在本地执行的程序开始，由浅入深，贯穿从 GCC 编译到 Make 与 Autotools 的使用，直至最后 Helloworld 的交叉编译，让学生直观地体验软件编译的工程实践，从实际角度理解编译原理，特别是交叉编译的原理。

## 二、 教法与学法

在教学手法上，主要采用以下方法来降低学习难度：

- 1) 循序渐进，由表及里，由浅入深——由于专科学生没有进行编译原理的理论学习，直接向学生介绍交叉编译的原理容易把知识“讲死”，学生还是无法理解跨处理器架构的交叉编译为什么要经过这么多复杂的步骤。而先从“Helloworld”的编译过程出发，通过课堂现场演示使用 gcc 生成编译过程各个阶段的中间结果，能够让学生直观地看到一个最简单程序的编译步骤，为学生理解交叉编译过程提供了一个“辅助的台阶”，从而让学习曲线不再那么陡峭。
- 2) 采用比较教学法——从学生熟悉的本地编译入手，通过比较本地编译和交叉编译的差别，让学生理解交叉编译中最关键的步骤：为什么需要一个临时的中间编译器作为过渡。理解了这一点，也就明白了交叉编译的实质，从而为

学生今后大量的交叉编译实践打下了坚实的理论基础。学生能够根据自己所理解的交叉编译过程去分析在实践中遇到的问题，起到“受人以渔”的效果。

- 3) “不求甚解”——传统的教学方法往往只注重知识的授予，而忽略的知识授予的顺序，亦即“知识的屏蔽”。在构建交叉编译环境的实践操作中，需要学生综合应用之前学到过的 Linux 用户环境设定、Shell 环境变量的运用、Autotools 的使用、内核头文件的使用、Make 工具的使用等等知识与技能，这就对学生掌握 Linux 系统相关知识的程度有了很高的要求，对于初次接触 Linux 系统的学生来说，这些要求是很难达到的。因此，在教学中，不应要求学生先理解每一行命令的含义，而是提供一个学生有能力自行完成的完整命令序列，让学生“不求甚解”，但求能够自己动手把交叉编译环境做出来。当学生通过自己努力后能够得到成果，再回过头来解释之前的操作步骤，学生的积极性能被调动起来，对理论内容的消化也变得相对容易。
- 4) 注重模仿——由于从源码实际构建交叉编译环境的步骤复杂，故在教学方法上采取“手把手”的方法：教师将每个操作步骤都记录下来并标注要点，让学生先立足于模仿得到结果，再进行知识点和技能点的拓展。
- 5) 理论与实践相结合，相辅相成——由于本课内容很大一部分是实际操作，而且，理论部分的讲解也需要通过教师实际操作的演示来帮助学生理解。故在课堂教学中，采取一边播放 PPT，一边在远程终端中通过命令行进行编译操作演示的教学方法，让学生“眼见为实”。

## 三、 教学程序

### 1. 新课导入

在导入新课时，先从学生熟悉的 Windows 系统和 VisualStudio 出发，通过对比 Windows 环境下集成开发环境的完善与易用，来凸显“纯手工”打造 Linux 嵌入式交叉编译环境的不易，激发学生的挑战精神和求知欲。同是，通过在本地编译“Hello world”的过程来设立一个参照系，为后续讲解中分解交叉编译的过程提供坐标。

### 2. 讲解

在讲解过程中，教师应作如下要求：

- 1) 注意观察编译中间结果的产生，通过中间结果的形式理解每个编译步骤所做的操作；
- 2) 随时提醒学生注意本地编译和交叉编译之间的联系与区别，必要时，通过提问学生引导学生自己进行总结；
- 3) 在进行交叉编译环境的构建实践时，要求学生先“按部就班”完成指令，再回头思考为什么这么做，观察并总结构建交叉编译工具链的实践操作与之前理论讲解的一致性。

通过教师对本地编译和交叉编译的对比讲解，以及教师对 gcc 编译过程、

Make 与 Autotools 使用的实际操作，配合学生对教师操作的模仿完成，能够让学生先看到成果再进行深化提高。

### 3. 教学效果的评价

本课通过以下提问和练习来获取教学效果的反馈：

- 1) 提问：
  - a) 请总结交叉编译和本地编译的相同点与不同点；
  - b) 在交叉编译中，关于平台的处理器架构，需要设定三个关键的变量：BUILD、HOST 与 TARGET，请解释这三者分别指的什么？如果在一台 x86 机器上编译能够在 ARM 平台上运行的“Helloworld”，BUILD、HOST 与 TARGET 分别应该取什么值？
  - c) 交叉编译过程中，为什么需要生成第一阶段的 gcc (1<sup>st</sup> Stage Gcc)，即这个临时的 gcc 有什么用？为什么不能直接生成最终的交叉编译 gcc？
- 2) 练习：学生使用教师提供的交叉编译环境源码与操作脚本，制作出自己的交叉编译 gcc，将作品提交给教师进行验证查看。

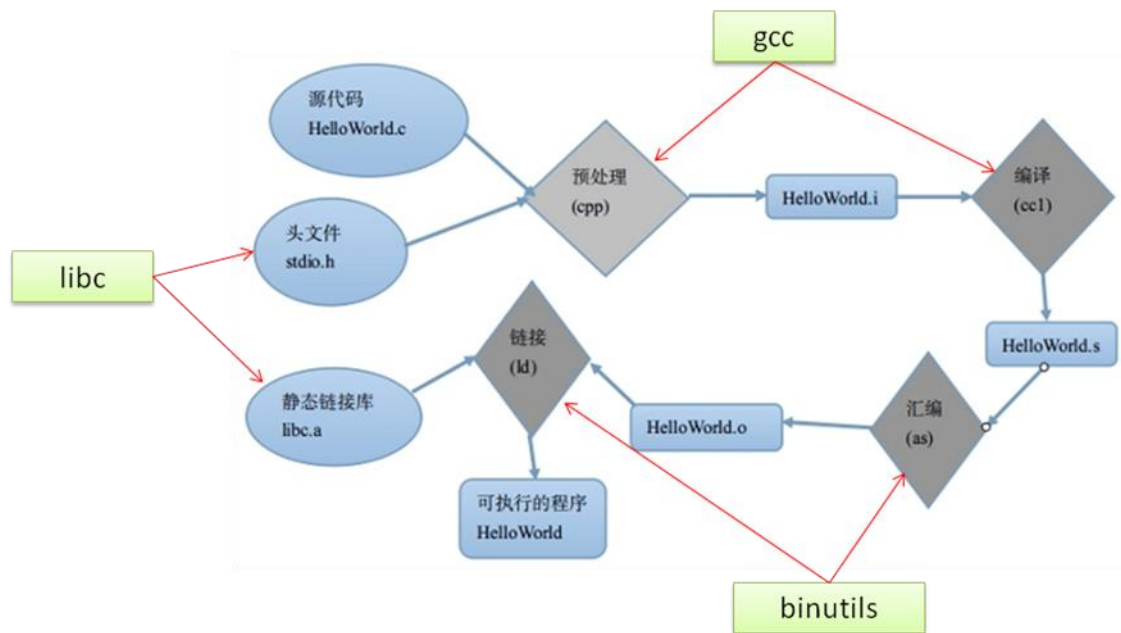
如果学生能够很好的回答上述提问，那么说明理论教学取得较好的效果；如果学生能够得到自己制作的交叉编译 gcc 程序，那么实践教学环节取得了较好的效果。

### 4. 归纳总结

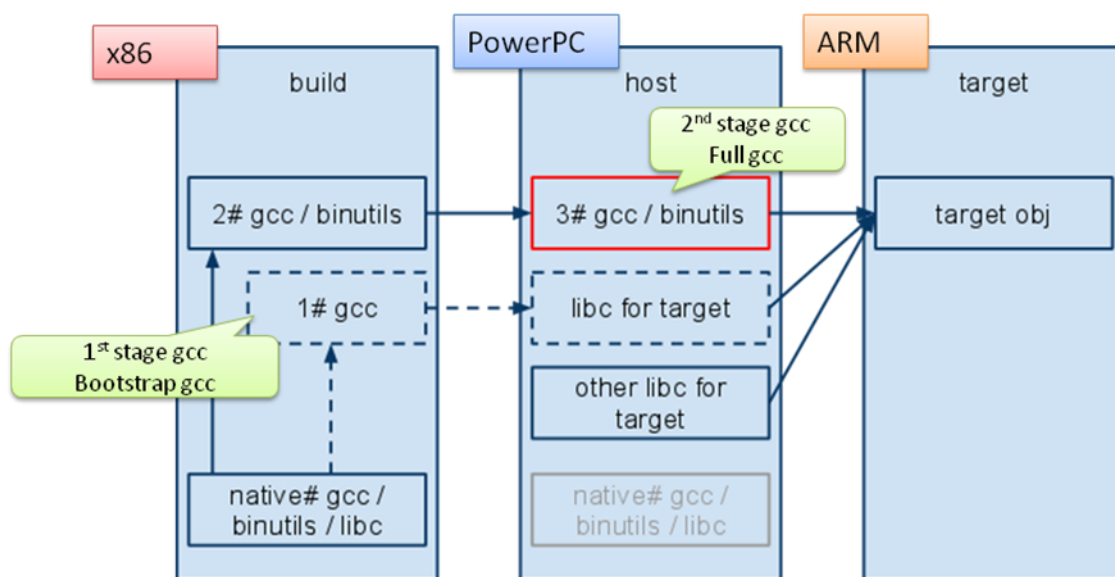
在 Linux 开源环境中针对 ARM 平台进行交叉编译环境的构建，是对本课程前面章节知识的即学即用以及为后续课程实践环节打下坚实的基础，在整个课程围绕 ARM 平台使用 GNU/Linux 软件环境构建嵌入式系统的知识体系结构中，本节课的内容是承前启后，既涉及到 ARM 处理器的硬件特性（对编译器有影响的特性，例如浮点单元），又涉及到 GNU/Linux 编译工具链的使用，其知识点涉及到前后课程，甚至是其他专业课的地方非常多，故重要性不言而喻，难度也堪称本门课程之最，故需要学生课上积极思考，课后勤于练习，方能真正掌握使用 Linux 开源技术构建嵌入式交叉编译环境的方法

## 四、 板书设计

由于本课知识点之间的关系错综复杂，并且实践性特别强，故多采用图的形式来梳理知识点联系以及让操作步骤及步骤间逻辑关系变得直观易懂



## GNU/Linux GCC工具链的交叉编译



- 交叉编译中 **build**、**host**、**target** 的概念
- 交叉编译中的依赖关系
- 交叉编译的过程

# 交叉编译中的组件依赖关系

