华清远见 FARSIGHT 嵌入式培训专家

嵌入式学院 华清远见旗下品牌

基于ARM平台的移植概述

潘友华

版权

- 华清远见嵌入式培训中心版权所有；
- 未经华清远见明确许可，不能为任何目的以任何形式复制或传播此文档的任何部分；
- 本文档包含的信息如有更改，恕不另行通知；
- 保留所有权利。

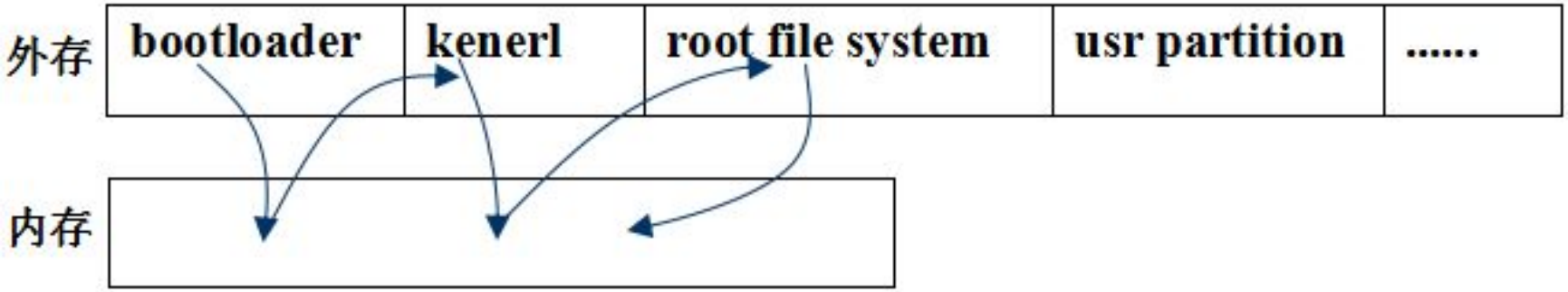
内容提纲

- 1、移植概述
  - 1.1、系统引导过程
  - 1.2、移植涉及内容
  - 1.3、移植方法概述
- 2、U-BOOT移植
  - 2.1、ARM平台引导方式
  - 2.2、两个阶段代码
  - 2.3、内核启动条件
  - 2.4、移植方法概述
- 3、内核移植
- 4、根文件系统制作
  - 4.1、概述
  - 4.2、内容
  - 4.3、格式
  - 4.4、烧写

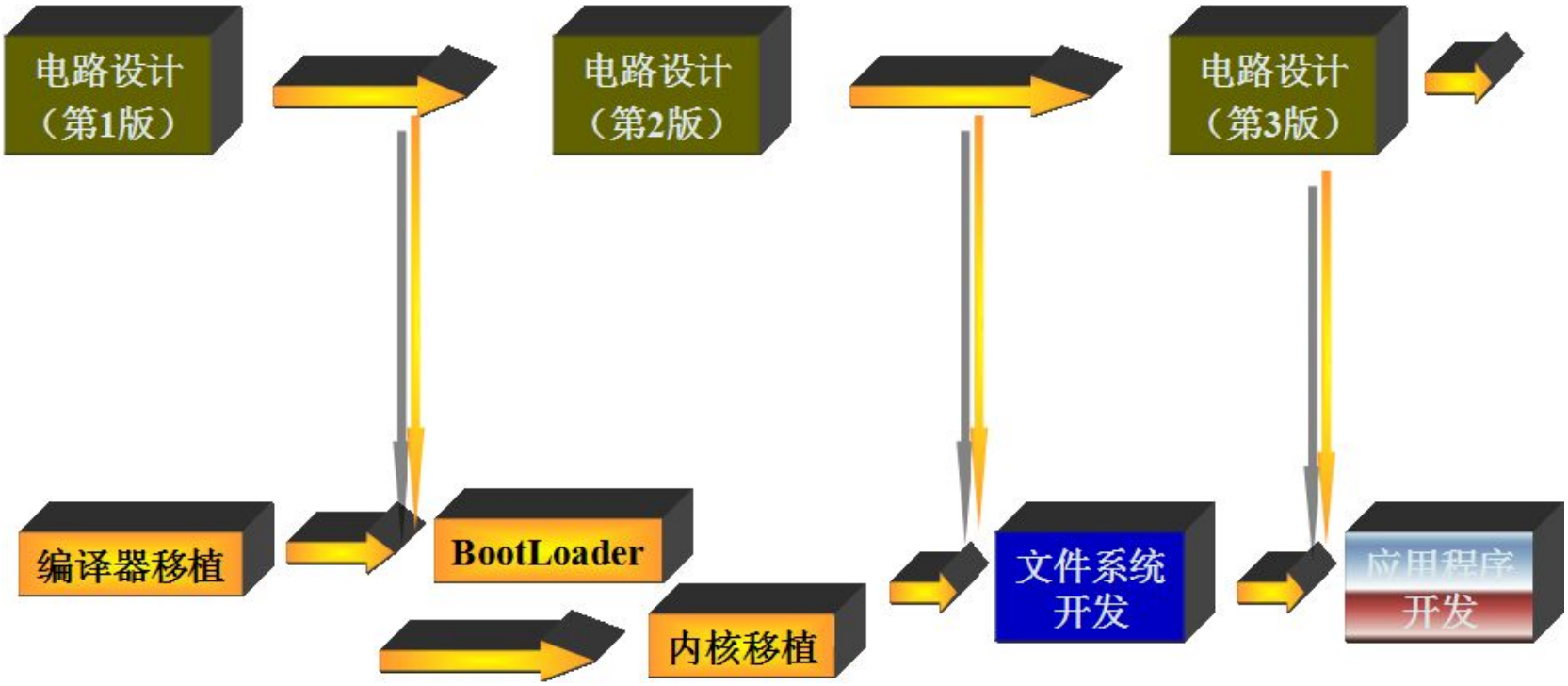
移植方法概述-系统引导过程

Target (ARM, PPC, etc.)

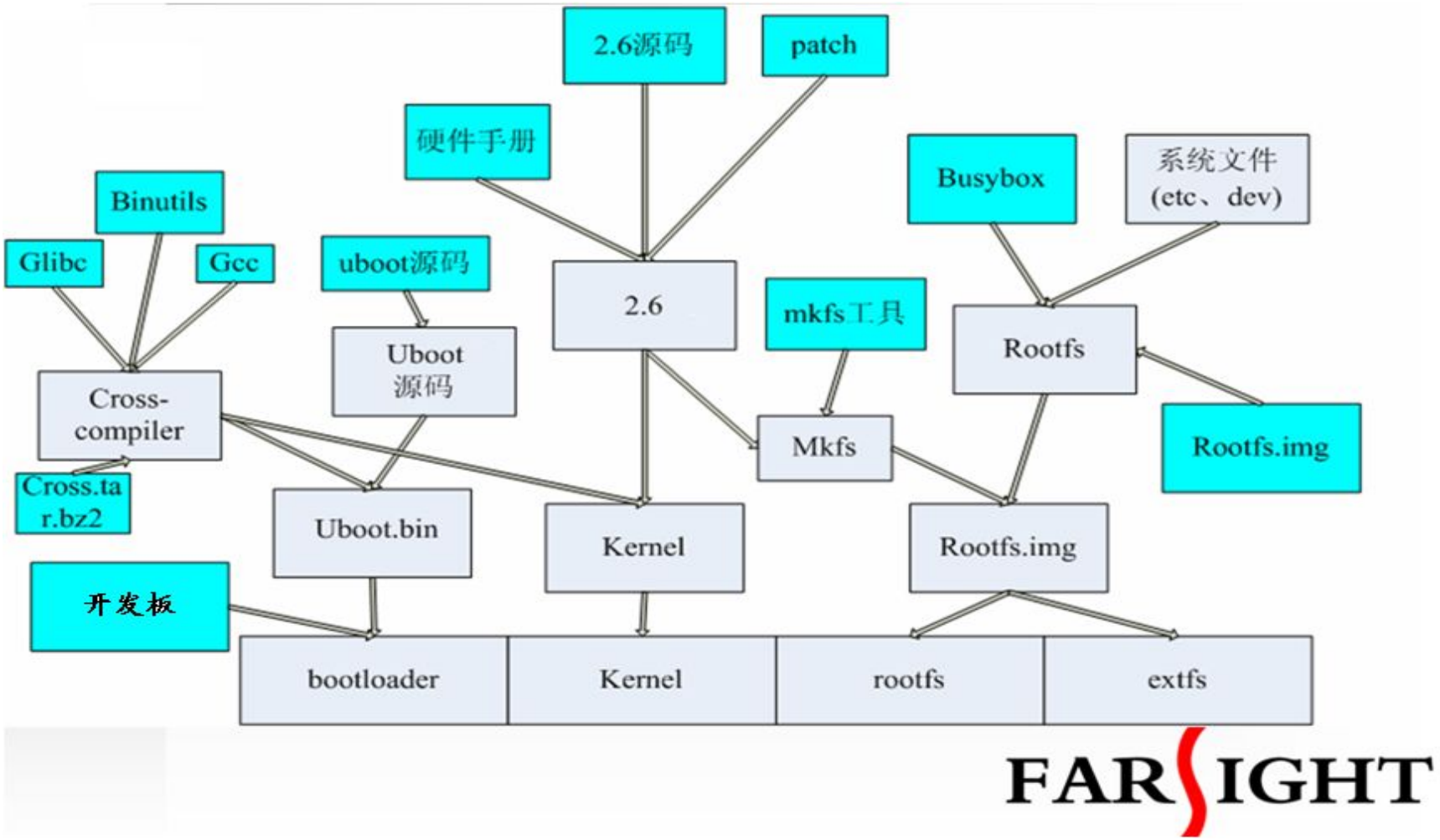
bootloader —初始化硬件/加载内核→ kernel —初始化/挂载rootfs→ application



移植方法概述-移植涉及内容



移植方法概述-移植涉及内容



移植方法概述-移植方法概述

- 一般的，基于成熟代码的移植，主要会有三个方面的工作：
  - A、有现成直接可用的代码，则选中编译
    - 主要是熟悉源码组织结构和编译系统。一般的，基于linux环境下的编译系统，一般都是由脚本及Makefile组成。
  - B、有现成的代码，但是部分代码不合适，则修改后选中编译
    - 除熟悉源码组织结构和编译系统外，还需要熟悉系统初始化流程，理解代码组织框架等等。
  - C、没有的代码，则编写添加
    - 一般的，需要添加的都是涉及设备控制的逻辑和功能模块。所以除熟悉源码组织结构和编译系统外，对该系统源码设计的思路要充分理解，按照其设计思路设计新的代码并加入到源码中。

U-BOOT移植-ARM平台引导方式

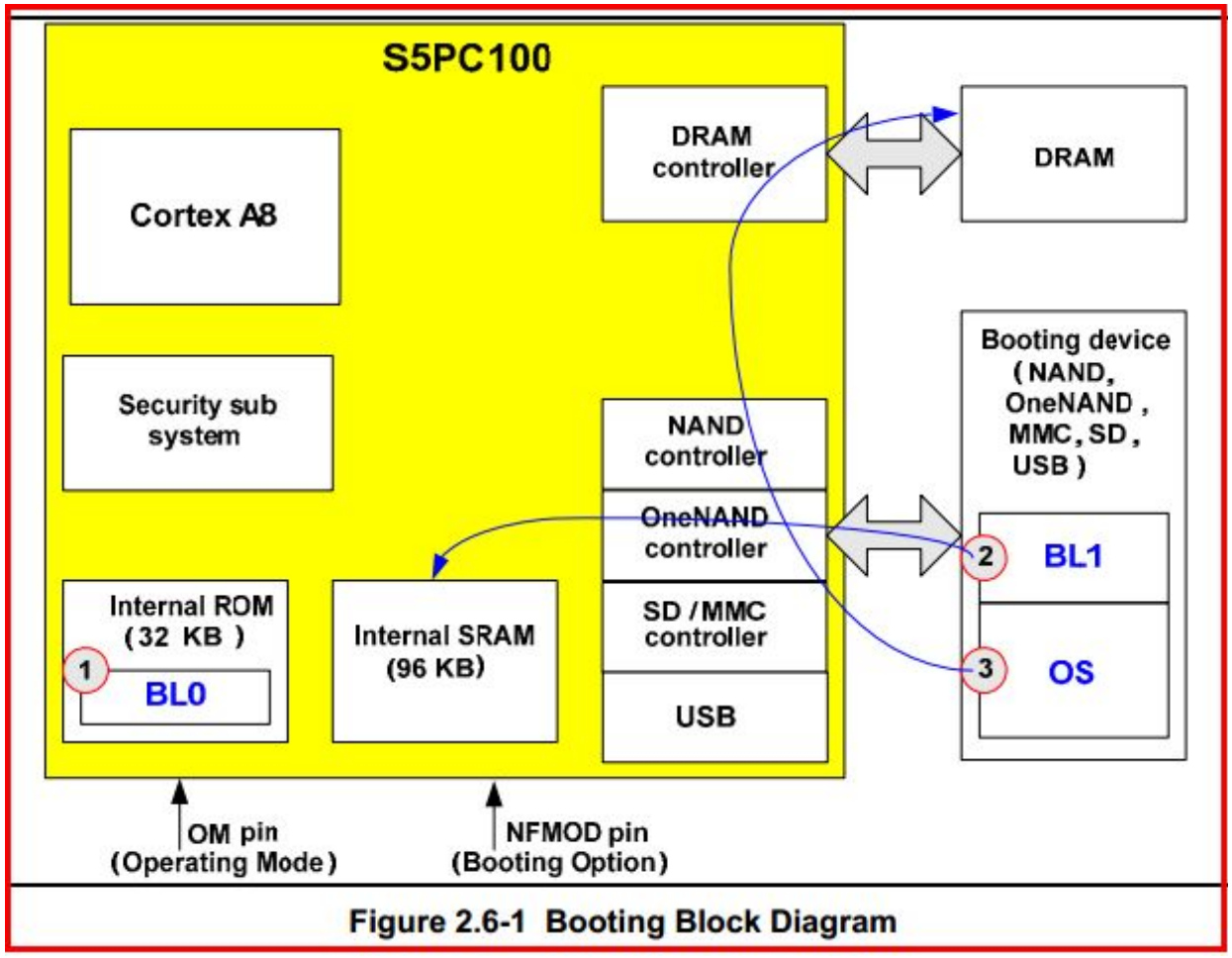
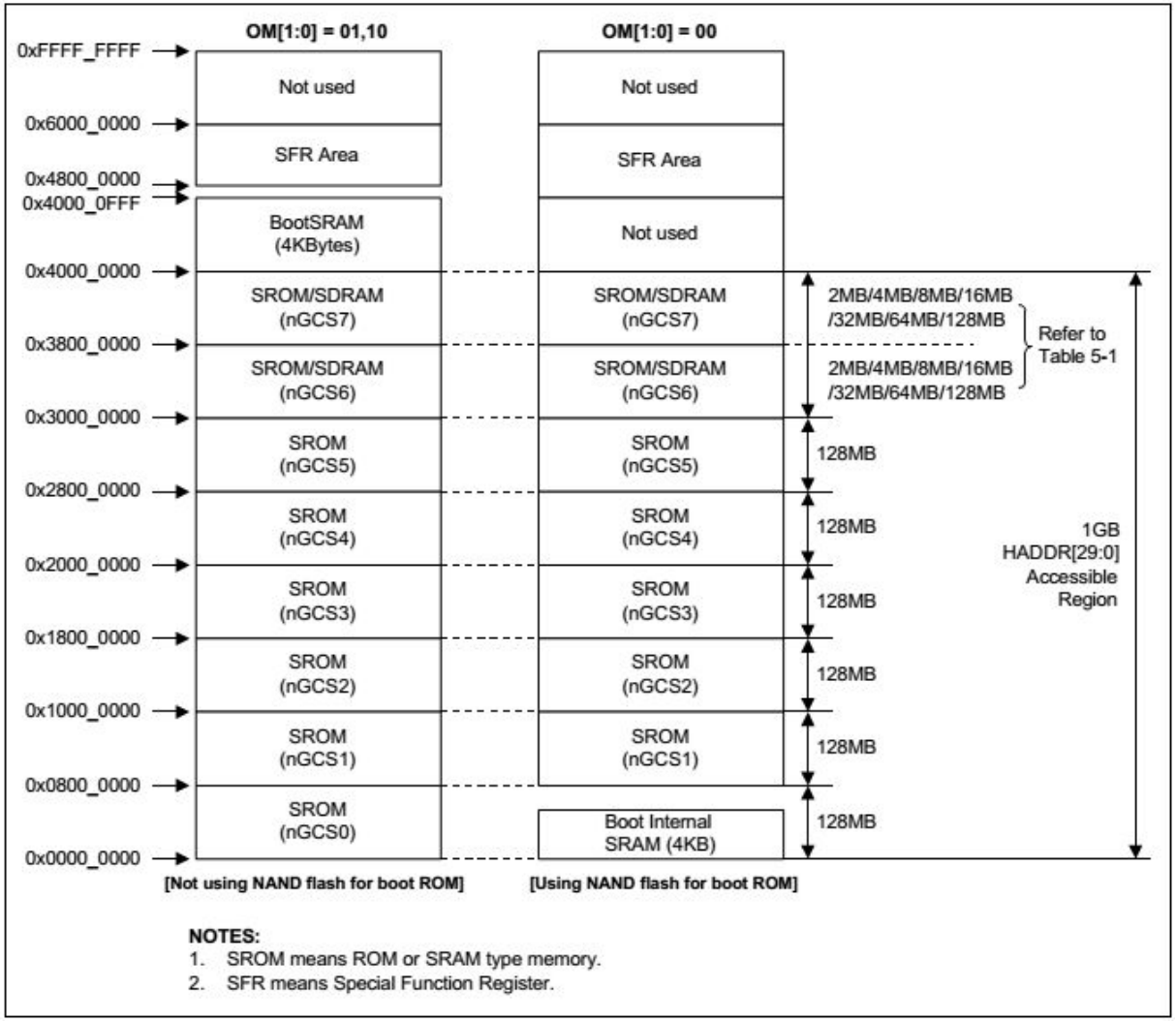


Figure 2.6-1 Booting Block Diagram

U-BOOT移植-ARM平台引导方式



U-BOOT移植-两个阶段代码

- 两个阶段的代码
- 第一阶段
  - 初始化基本的硬件
  - 把bootloader自搬运到内存中
  - 设置堆栈指针并将*bss*段清零。为后续执行C代码做准备
  - 跳转到第二阶段代码中
- 第二阶段
  - 初始化本阶段要使用到的硬件
  - 读取环境变量
  - 如果是自启动模式，从Flash或通过网络加载内核并执行
  - 如果是下载模式，接收到用户的命令后执行

U-BOOT移植-内核启动条件

- 内核启动的准备条件：
  - A、保存平台号到r1中
  - B、保存参数地址到r2中(param_struct 或 taglist)
  - C、关闭 d-cache i-cache mmu 等

U-BOOT移植-移植方法概述

- 三个方面的工作：
  - 1、选平台
    - A、指定交叉编译工具链
    - （1）、修改顶层Makefile
    - ifeq (arm,$(ARCH))
    - CROSS_COMPILE ?=arm-linux-
    - endif
    - （2）、编译指定
    - # make ARCH=arm CROSS_COMPILE=arm-linux-
    - B、添加缺省配置
    - （1）、修改顶层Makefile
    - 添加
    - xxx_config: unconfig
    - @$(MKCONFIG) xxx arm 芯片 板子名 公司名 soc
    - （2）、修改boards.cfg
    - xxx arm 芯片名 板子名 公司 soc
    - 本质是在指定编译的平台相关的目录

U-BOOT移植-移植方法概述

- 2、定制/修改板级文件
  - 板级文件主要做了如下工作：
  - A、参数指定
  - 机器码、内核启动参数地址等等
  - B、板级初始化
  - 设备主机端初始化及设备参数
  - C、低级初始化代码
  - 内存初始化、时钟、看门狗等等
- 板级文件
  - \board\公司名\板子名\
    - lowlevel_init.S
    - 板级文件
    - 外存读写
    - 内存初始化

U-BOOT移植-移植方法概述

- 3、模块选配
  - A、命令
  - B、逻辑模块
  - C、驱动
- 选配方法
  - include\configs\xxx.h
  - 这个头文件中主要有如下内容：
  - A、参数
  - B、开关
  - C、参数开关
  - 宏开关出处
    - 相应源码目录下的Makefile

内核移植

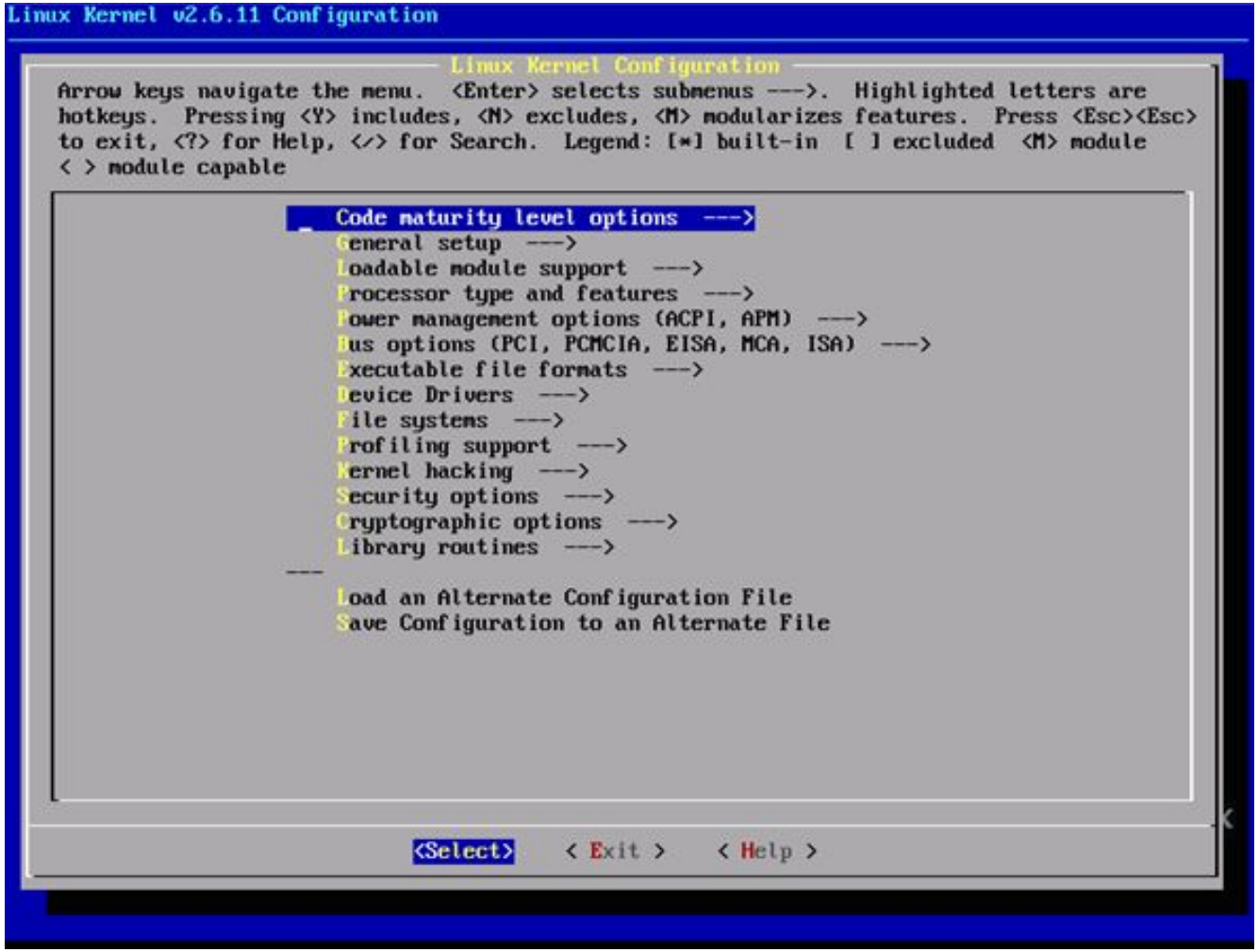
- 1、选平台
  - A、指定交叉编译工具链
  - （1）、修改顶层Makefile
    - ARCH ?= arm
    - CROSS_COMPILE ?= arm-linux-
  - （2）、编译指定
    - # make ARCH=arm CROSS_COMPILE=arm-linux-
  - B、拷贝arch\arm\configs\xxx_defconfig到顶层目录下
    - # make xxx_defconfig
  - C、编译验证

内核移植

- 2、定制/修改板级文件
  - 板级文件主要做了如下工作：
  - A、参数指定
  - 机器码、内核启动参数地址、物理I/O的起始地址等等
  - B、板级初始化
  - 时钟、中断、I/O映射、串口等等初始化工作
  - C、设备主机端初始化及设备参数
- 板级文件
  - arch/arm/mach-xxx/xxx.c
  - MACHINE_START(XXX, "XXX")
  - .phys_io = ...,
  - .io_pg_offst = ...,
  - .boot_params = ...,
  - .init_irq =....,
  - .map_io =....,
  - .init_machine =....,
  - .timer = ...,
  - MACHINE_END

内核移植

- 3、模块选配
  - A、逻辑模块
  - B、驱动



根文件系统制作-概述

- 文件系统就是管理外存上数据的逻辑结构，是存储、组织计算机文件和数据的一种方法，更严格地说，是一套实现了数据的存储、分级组织、访问和获取等操作的抽象数据类型(Abstract data type)，通过它来访问、查找文件和数据很容易。外存数据在硬盘就是具体位置，用代码表述的话就是起始地址，但是看着不直观，操作也不方便，有文件系统弄出文件及目录的概念来翻译转换这些数据区域操作更方便。打个比方：一本书中就好比文件系统，目录及内容就是被文件系统管理的正文，有了目录，读写查找书的内容才方便。
- 涉及概念：
  - A、内容
  - 命令、脚本、配置、目录等等
  - B、格式
  - 文件系统的内容总是以特定格式存储在外存上的。嵌入式设备中常用flash作为外存，对应的文件系统有yaffs/yaffs2、jffs2、cramfs等等。

根文件系统制作-内容

- 根文件系统常见内容
  - A、命令
    - bin/*、sbin/*、usr/bin/*、usr/sbin/*
    - init or linuxrc
  - B、脚本、配置
    - etc/*
    - 初始化脚本、系统及应用程序配置文件、服务及命令等脚本
  - C、设备文件
    - dev/*
    - 手工创建，系统自动生成（mdev/udev）
  - D、共享库
    - lib/*
  - E、系统目录
    - sys/*　　设备信息
    - proc/*　　系统系统
    - tmp/*　　临时文件
    - var/*　　日志文件

根文件系统制作-格式

- 根文件系统常见格式
- YAFFS（Yet Another Flash File System）
  - 是第一个专门为NAND Flash存储器设计的嵌入式文件系统，适用于大容量的存储设备；并且是在GPL（General Public License）协议下发布的，可在其网站免费获得源代码。是基于日志的文件系统，提供磨损平衡和掉电恢复的健壮性。它还为大容量的Flash 芯片做了很好的调整，针对启动时间和RAM 的使用做了优化。它适用于大容量的存储设备，已经在Linux 和WinCE 商业产品中使用。
- ramfs（ram-based filesystem）
  - ramfs是一个非常简单的文件系统，顾名思义是内存文件系统，它它处于虚拟文件系统（VFS）层，而不像ramdisk那样基于虚拟在内存中的其他文件系统(ex2fs）。
- JFFS（Journalling Flash File System）
  - 闪存设备日志型文件系统最初是由瑞典的 Axis Communication AB 开发，其目的是作为嵌入式系统免受宕(dang)机和断电危害的文件系统。然而用于NAND设备上JFFS已被JFFS2大量取代。

根文件系统制作-烧写

- 烧写根文件系统
- A、手工制作镜像包
  - （1）使用专用工具把根文件系统的内容按指定格式制作成一个镜像包
  - （2）使用u-boot这样的bootloader烧写该镜像包
- B、挂载拷贝
  - （1）烧写分区以指定格式挂载到指定目录
  - （2）向该目录拷贝根文件系统的内容

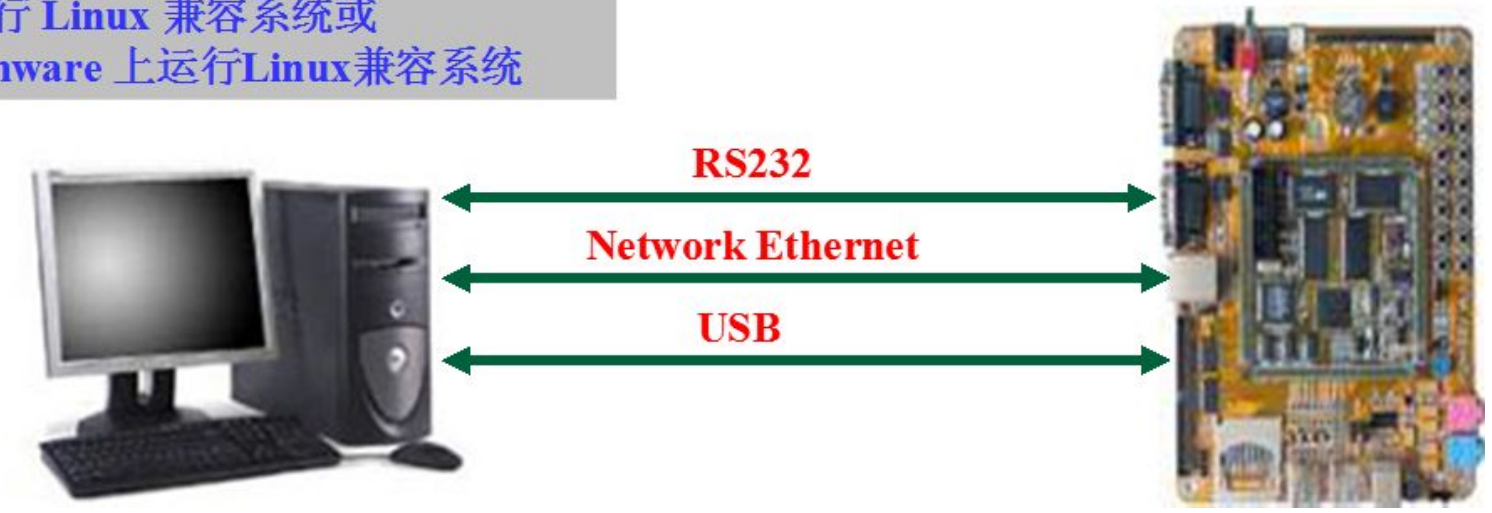
根文件系统制作-烧写

开发主机（Host）

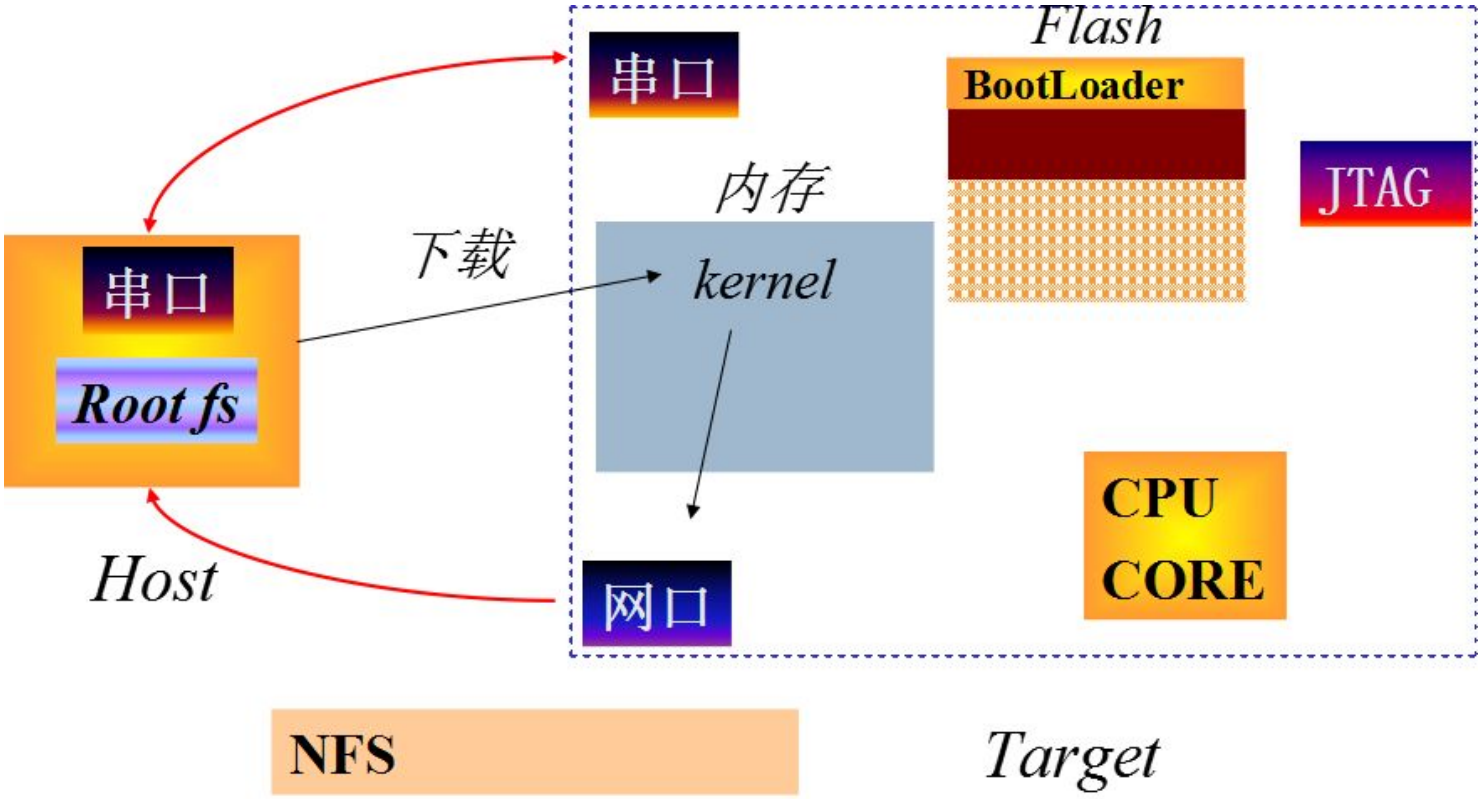
目标机（Target）

连接介质

运行 Linux 兼容系统或
Vmware 上运行Linux兼容系统



根文件系统制作-烧写


Flash
串口
BootLoader
JTAG
内存
下载
kernel
串口
Root fs
CPU
CORE
Host
网口
NFS
Target