



ARM+Linux嵌入式系统技术路线

www.farsight.com.cn

- ✓ ARM + Linux嵌入式开发背景
- ✓ 嵌入式Linux系统开发技术路线
 - ∅ 交叉编译环境
 - ∅ Bootloader
 - ∅ Linux系统移植
 - ∅ 文件系统
 - ∅ Linux应用程序的开发
- ✓ Linux系统开发模式

ARM + Linux 开发背景

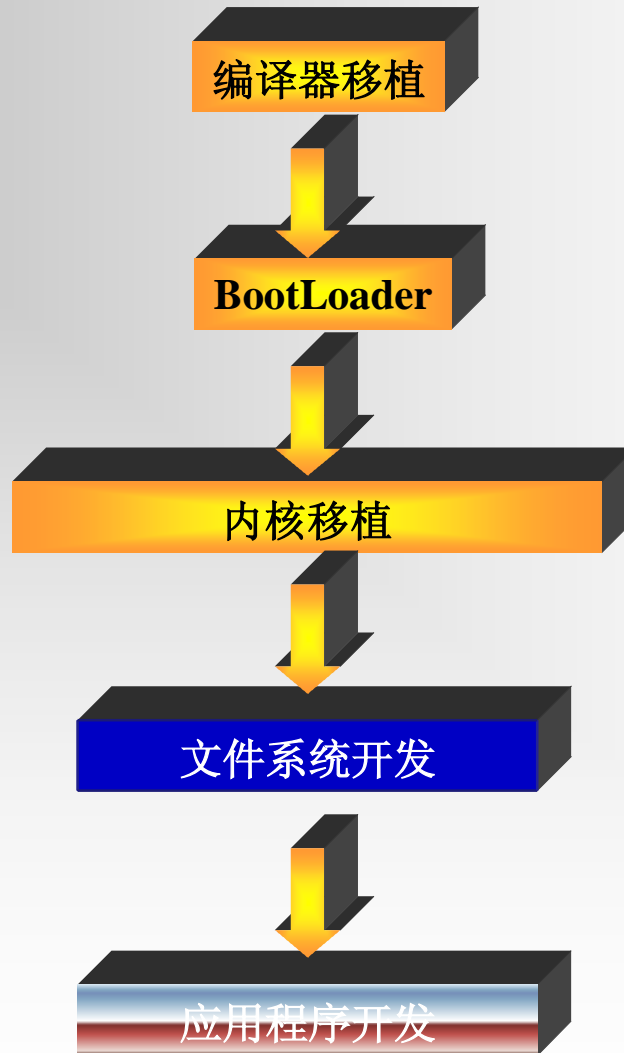
√ 与传统Windows开发的差异

- ∅ Windows下编程 vs Linux下编程 (语言, IDE等)
- ∅ Windows下工具 vs Linux下工具 (仿真器等)
- ∅ 操作系统 (一般嵌入式操作系统, Linux操作系统)
- ∅ 开发者的熟悉程度

√ 嵌入式Linux开发要点

- ∅ 选择嵌入式Linux版本
- ∅ 熟悉开发环境和工具
- ∅ 熟悉Linux内核
- ∅ 熟悉目标板引导方式
- ∅ 根文件系统
- ∅ Linux调度机制和进程模型

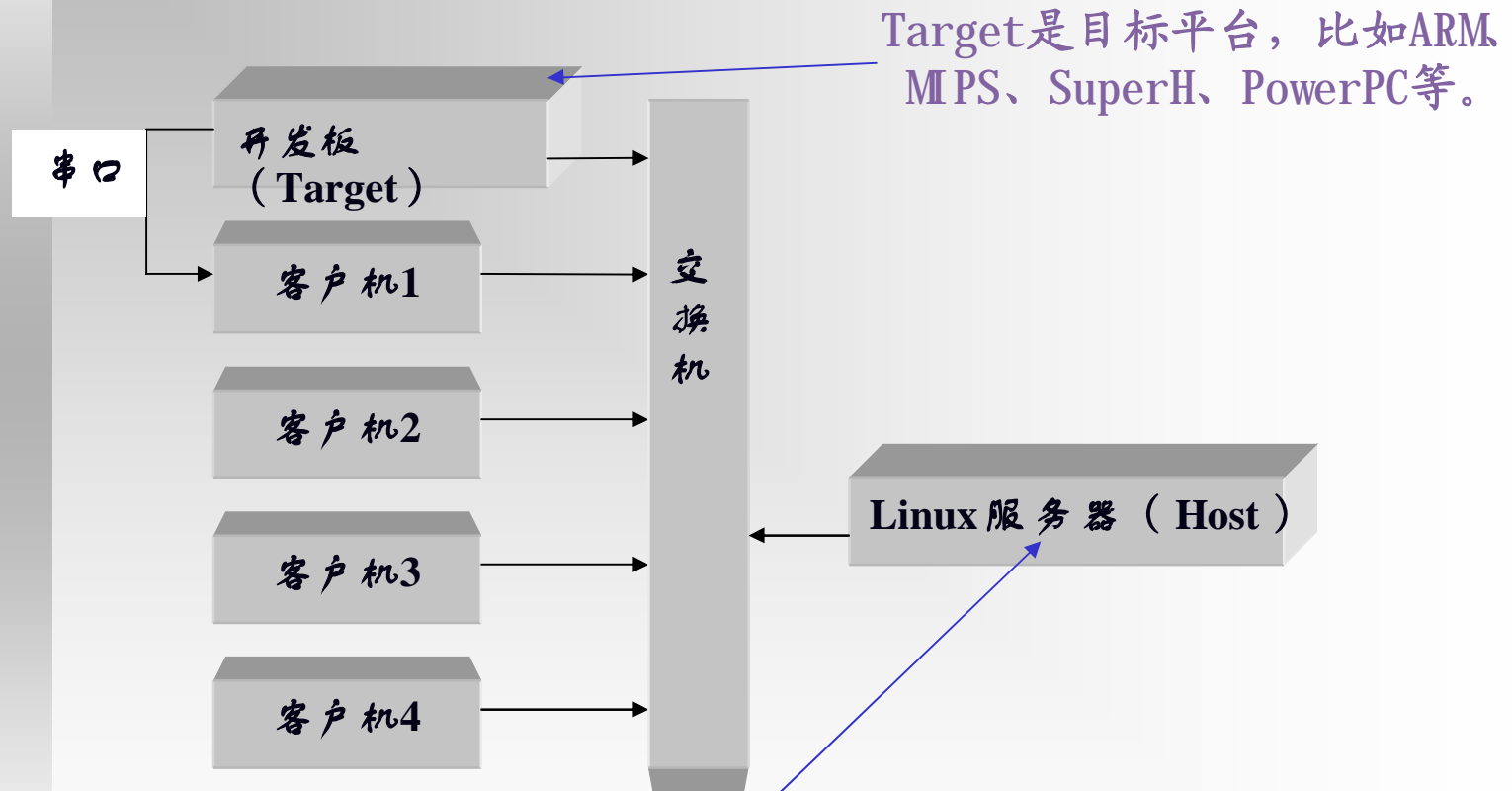
嵌入式linux系统开发技术路线



✓ 在开始嵌入式项目之前，必须具备Linux编程的基本技能：

- ∅ Linux下C语言编程，调试
- ∅ 基本Linux命令的使用
- ∅ 基本Linux工具如minicom, FTP, TFTP等
- ∅ Makefile规则
- ∅

√ 主机与目标板开发模式



Target是目标平台，比如ARM、MIPS、SuperH、PowerPC等。

是一个基于x86架构的计算机（Pentium4处理器、Athlon处理器等），即i386平台。

GCC 交叉编译器

√ GNU Binutils

∅ GNU Binutils 的主要工具有两个，一个是连接程序ld，另外一个汇编程序as。其主要目的是为GNU 系统，提供汇编和连接工具。

√ GNU GCC

∅ GNU GCC 就是上面提到的GCC，GCC 主要是为GNU 系统提供C 编译器。现在支持多种语言，这其中包括C/C++、Fortran、Java、Objective-C、甚至还有Ada。

√ GNU GLibc

∅ 用于定义系统调用和其它一些基本的函数调用。

GCC交叉编译器的制作

- ✓ 第1步，取得Binutils、GCC、Glibc 源码。
- ✓ 第2步，配置并编译Binutils 取得我们所需要的汇编和连接程序。
- ✓ 第3步，配置并编译GCC 源码生成GCC 编译器。一般是C 编译器首先生成，然后以这个为基础在结合下一步生成的Glibc 的C 函数库，再编译生成其它编译器。
- ✓ 第4步，配置Glibc 并编译生成Glibc 的C 函数库。
- ✓ 第5步，再次配置和编译GCC 源码，生成其它语言的编译器，如C++编译器。

设置环境变量

```
export PROJECT=S3C2410TK
export PRJROOT=~/$PROJECT
export TARGET=arm-linux
export PREFIX=$PROJECT/tools
export TARGET_PREFIX=$PREFIX/$TARGET
export PATH=$PREFIX/bin:$PATH
```

交叉编译环境的快速建立

- ✓ 下载 gcc软件包
- ✓ 执行解压命令
- ✓ 配置环境变量

• 交叉编译环境非常重要!

√ Bootloader介绍

- ∅ 计算机或嵌入式系统从开机到操作系统启动的过程

√ Bootloader启动

- ∅ 网络启动方式（文件系统在服务器）
- ∅ Flash启动方式（nand flash, Linux内核以及文件系统都存储在flash介质中）

u-boot简介

- ✓ U-boot是在PPC-Boot基础上进化而来的一个开发源码的BootROM程序。
- ✓ 采用了高度模块化的编程方式

u-boot启动过程

- ✓ `_start:` `--> reset:`
 - `--> cpu_init_crit` `--> memsetup`
 - `--> relocate:`
 - `--> stack_setup:`
 - `--> start_armboot()` `-->`
`init_sequence[]`
- ✓ `--> ...`
- ✓ `--> main_loop()`

u-boot执行go（或者bootm）命令，引导Linux内核。

✓ 配置 u-boot

∅ make <BOARD_NAME>_config
make smdk2410_config

✓ Building

∅ make CROSS_COMPILE = arm-linux-

✓ 生成的文件:

∅ <i>System.map</i>	The symbol map
∅ <i>u-boot</i>	U-Boot in ELF
∅ <i>u-boot.bin</i>	U-Boot raw binary image
∅ <i>u-boot.srec</i>	U-Boot image in Motorola's S-Record format

移植u-boot的步骤

- ✓ 下载源代码
 - ✓ 阅读Readme文件
 - ✓ 了解u-boot结构
 - ✓ 添加修改
 - ✓ 编译
 - ✓ 调试
 - ✓ 发布
- ✓ include/configs/smdk2410.h
 - ✓ board/smdk2410/smdk2410.c
 - Ø int board_int(void)
 - Ø int dram_init(void)
 - ✓ board/smdk2410/flash.c
 - ✓ cpu/arm920t/serial.c
 - ✓ drivers/rtl8019.c

烧写 Bootloader

- ✓ 需要对Flash进行烧写，固化boot-loader
 - ∅ 利用flash烧写器
 - ∅ 利用jtag工具(+sjf2410)

√ Linux内核下载

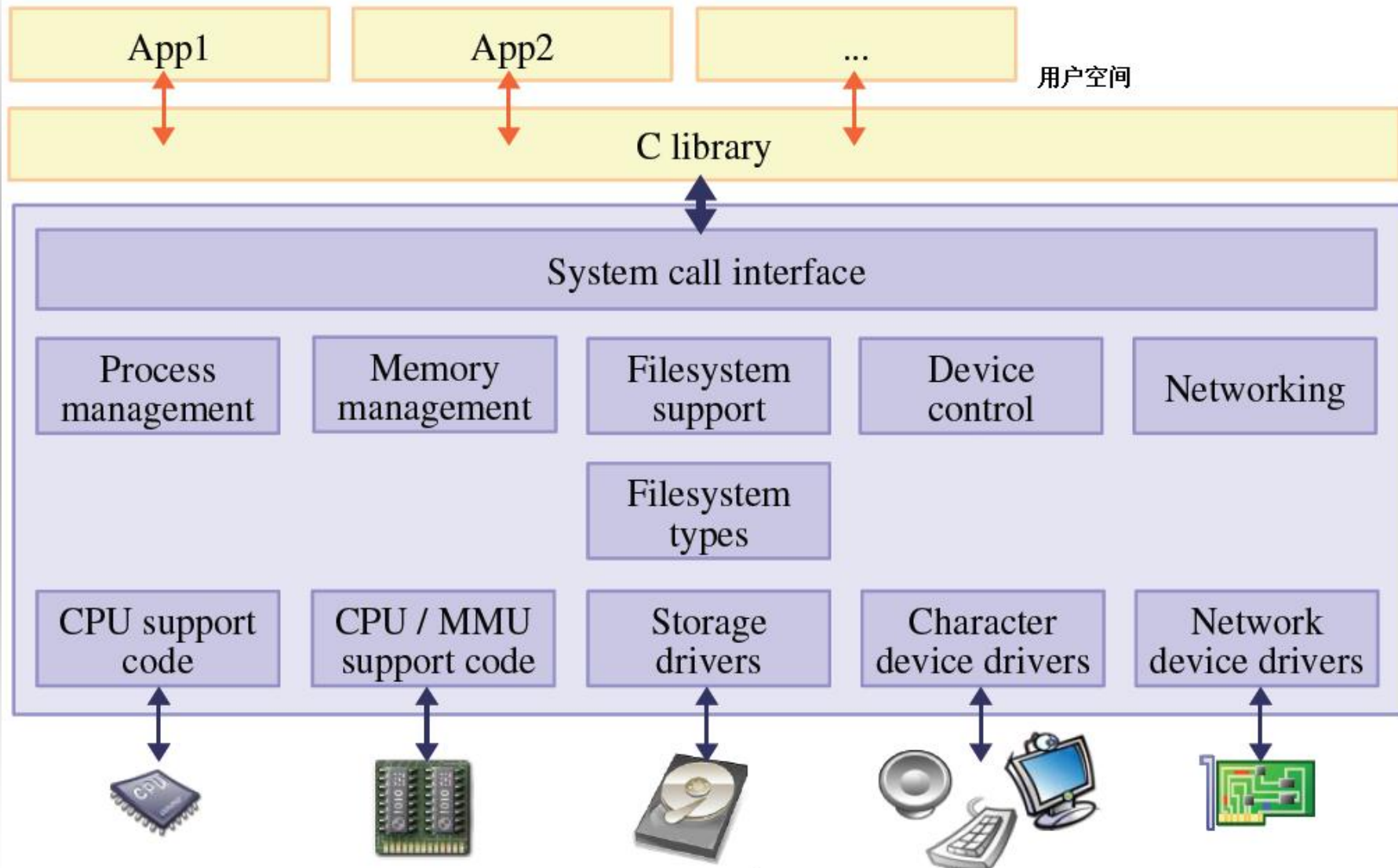
Ø <http://kernel.org>

Ø wget <http://kernel.org/pub/linux/kernel/...>

Ø wget http://kernel.org/.../.../linux-*.bz2.sign

Ø gpg -verify linux-2.xx.tar.bz2.sign

Ø tar jxvf linux-2.xx.tar.bz2



✓ 进入到内核源代码目录

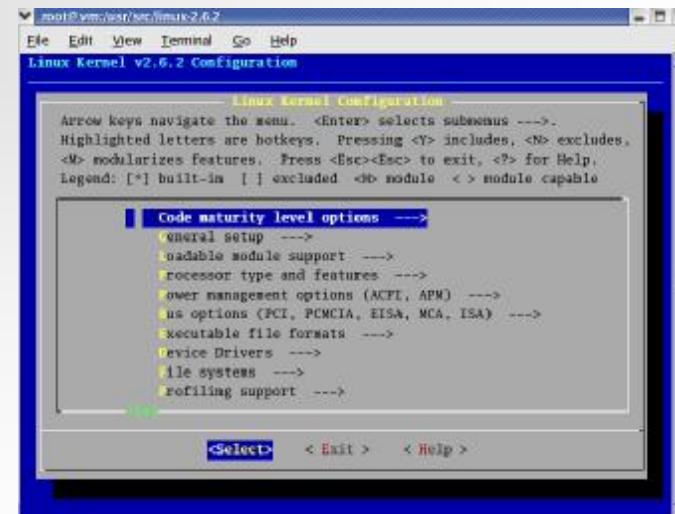
Ø `cd /usr/src/linux-2.6.15`

✓ `make config`

✓ `make xconfig`

✓ `make menuconfig`

✓ `make`



✓ 修改目标平台

∅ 例如：ARM 平台，crosscompiler command:
arm-linux-gcc

∅ ARCH = arm

∅ CROSS_COMPILE = arm-linux-

∅ The Makefile defines later CC =
\$(CROSS_COMPILE)gcc

✓ 或运行下面命令

∅ make ARCH=arm CROSS_COMPILE=arm-linux-

移植Linux内核

- ✓ arch/arm/config.in
- ✓ arch/arm/Makefile
- ✓ arch/arm/mach-s3c2410/*
- ✓ include/asm-arm/arch-s3c2410/*
- ✓ arch/arm/kernel/debug-armv.S
- ✓ arch/arm/kernel/entry-armv.S
- ✓ arch/arm/kernel/head-armv.S
- ✓ arch/arm/tools/mach-types
- ✓ arch/arm/boot/compressed/head-s3c2410.S
- ✓ arch/arm/boot/compressed/Makefile
- ✓ arch/arm/boot/Makefile

技术路线之 文件系统

✓ 文件系统 (FileSystem) 是一组程序，它们告诉操作系统如何访问及解释存储在磁盘或磁带驱动器或者其它存储媒介上的内容

✓ Cramfs

- ∅ Linus Torvalds 开发
- ∅ cramfs是只读压缩的文件系统，文件系统类型可以是ext2, ext3
- ∅ 系统启动后，Linux kernel把cramfs文件系统载入到内存中，解压，所以比较占内存
- ∅ 采取了 zlib 做压缩,省空间

√ Yaffs

- ∅ YAFFS, Yet Another Flash File System, 是一种类似于 JFFS/JFFS2 的专门为 Flash 设计的嵌入式文件系统。
- ∅ 与 JFFS 相比, 它减少了一些功能, 因此速度更快、占用内存更少。

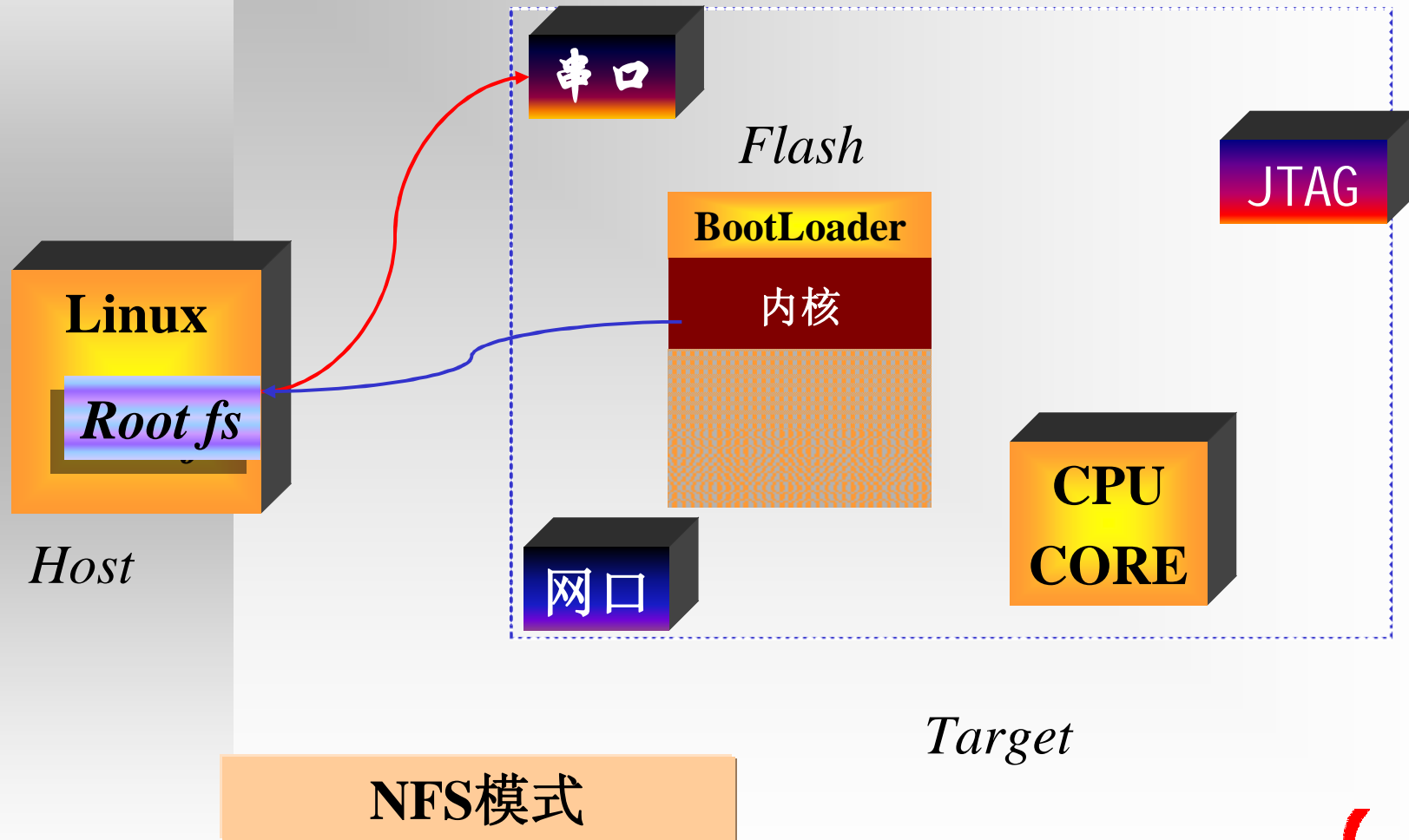
√ RAM Disk

- ∅ 如果使用 RAM disk, 一般应选择 EXT2 文件格式, 但 EXT2 并不是一块特别高效的文件存储空间。由于存在 RAM disk 上, 所以任何改变在下一次启动后都会丢失。当然, 也有许多人认为对嵌入式存储空间来讲, 这是一种优势, 因为每次系统启动都是从已知的文件系统状态开始的。
- ∅ Ramdisk 也是 Linux 系统中的一种虚拟设备, 是从内存中划分出来作为高速缓存的一部分, 将它虚拟为磁盘。Ramdisk device (如: /dev/ram0, /dev/ram1) 可以在任何时候被创建和加载, 同其他磁盘分区一样被操作。Ramdisk 通常用于系统启动。

技术路线之 应用程序的开发

- ✓ Linux应用程序的开发
- ✓ 在所有环境都搭建完毕之后，开始Linux的应用开发：
 - ∅ 文件IO编程
 - ∅ 进程控制开发
 - ∅ 进程间通讯
 - ∅ 多线程技术
 - ∅ 网络编程
 - ∅ GUI编程

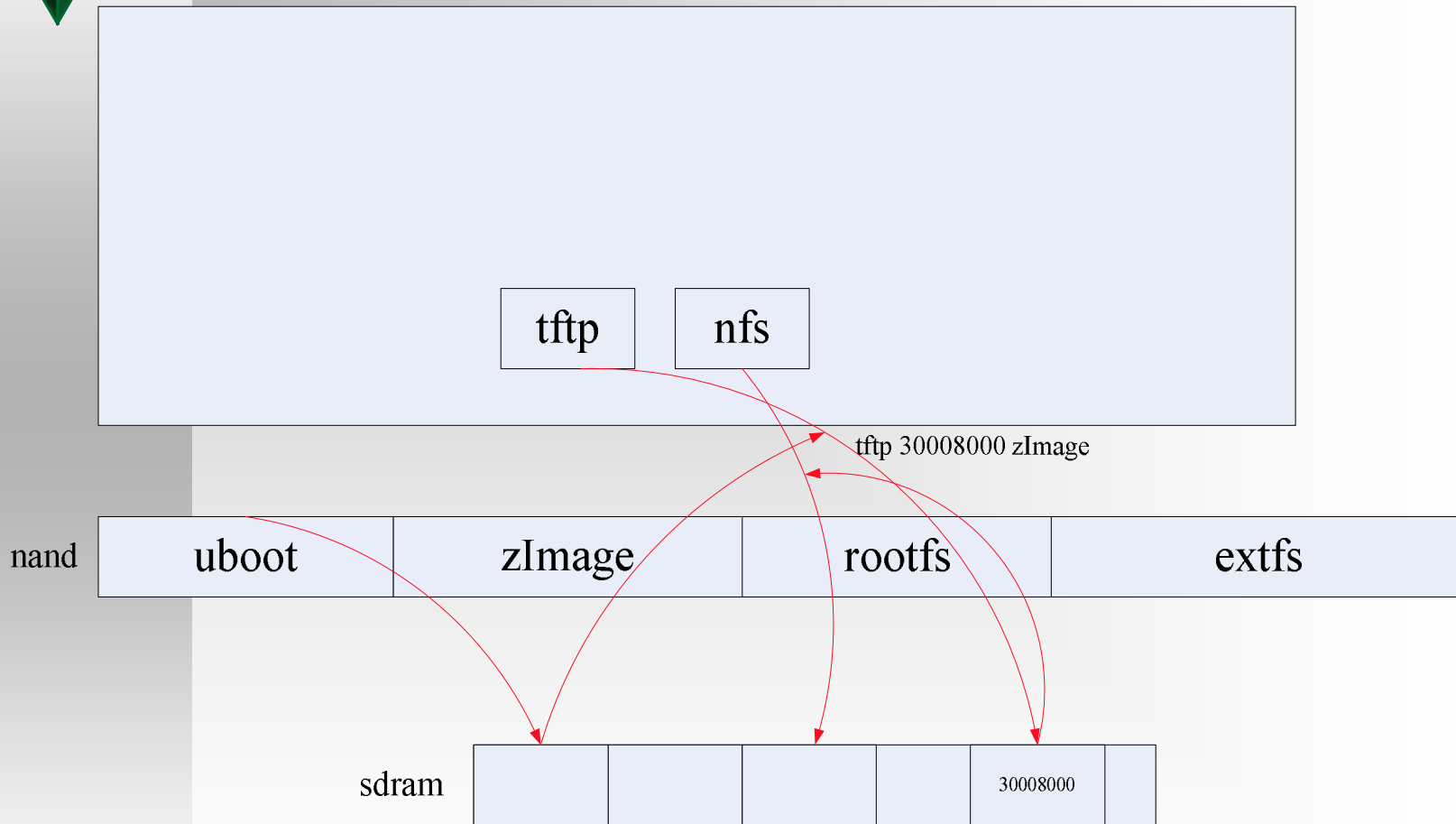
嵌入式Linux系统开发模式



NFS 开发模式

- ✓ 制作放入应用程序的rootfs
- ✓ 配置TFTP服务
- ✓ 设置NFS服务
- ✓ 利用uboot进行调试开发
- ✓ 调试完毕制作kernel和rootfs
- ✓ 烧写kernel

NFS 开发模式



- ✓ 我只关心系统移植，编程我很熟悉，学什么？
 - ∅ 系统移植的方法
 - ∅ 制作Bootloader
 - ∅ 编译调试内核
 - ∅ 定制文件系统

嵌入式Linux系统开发班 (FSEL3001)

- ✓ 学会简单的系统移植就行，创建出开发平台就行，编程不太会，怎么学？
 - ∅ 熟悉Linux操作系统具体操作
 - ∅ 熟悉并可以编译Linux内核
 - ∅ 熟悉嵌入式Linux系统开发的整体流程
 - ∅ 熟悉Linux系统调用能够熟练编写Linux应用程序，包括网络、图形等
 - ∅ 了解Linux设备驱动程序的开发

嵌入式Linux应用开发班 (FSLA1001)

- ✓ 系统移植和编程都比较熟悉了，但是硬件是自己设计的，怎么办？
 - ∅ Linux字符设备驱动程序
 - ∅ 内核内存管理与中断处理
 - ∅ 块设备驱动程序，MTD以及文件系统
 - ∅ 网络驱动程序
 - ∅ USB设备驱动程序开发
 - ∅ 帧缓冲驱动程序和DMA

嵌入式Linux设备驱动开发班 (FSLD3001)

✓ 我现在刚入门，都想学习一下，有这样的课程吗？

∅ 课时：1个月，总计176学时

∅ 讲课内容：基础篇、应用篇、移植篇、驱动篇

∅ 实验内容：

ü 个人媒体播放器（PMP）的实现

ü 商场POS机的实现

ü 智能车载导航仪GPS的实现

嵌入式Linux就业培训班（FSJY1001）

The logo features the words "FAR SIGHT" in white, uppercase, serif font. A red, stylized vertical bar separates the two words. The text is centered within a dark green, textured, inverted triangle that has a 3D effect with a white border.

FAR SIGHT

嵌入式培训专家

Thank you!

www.farsight.com.cn