



Android底层架构分析与移植

华清远见

版权

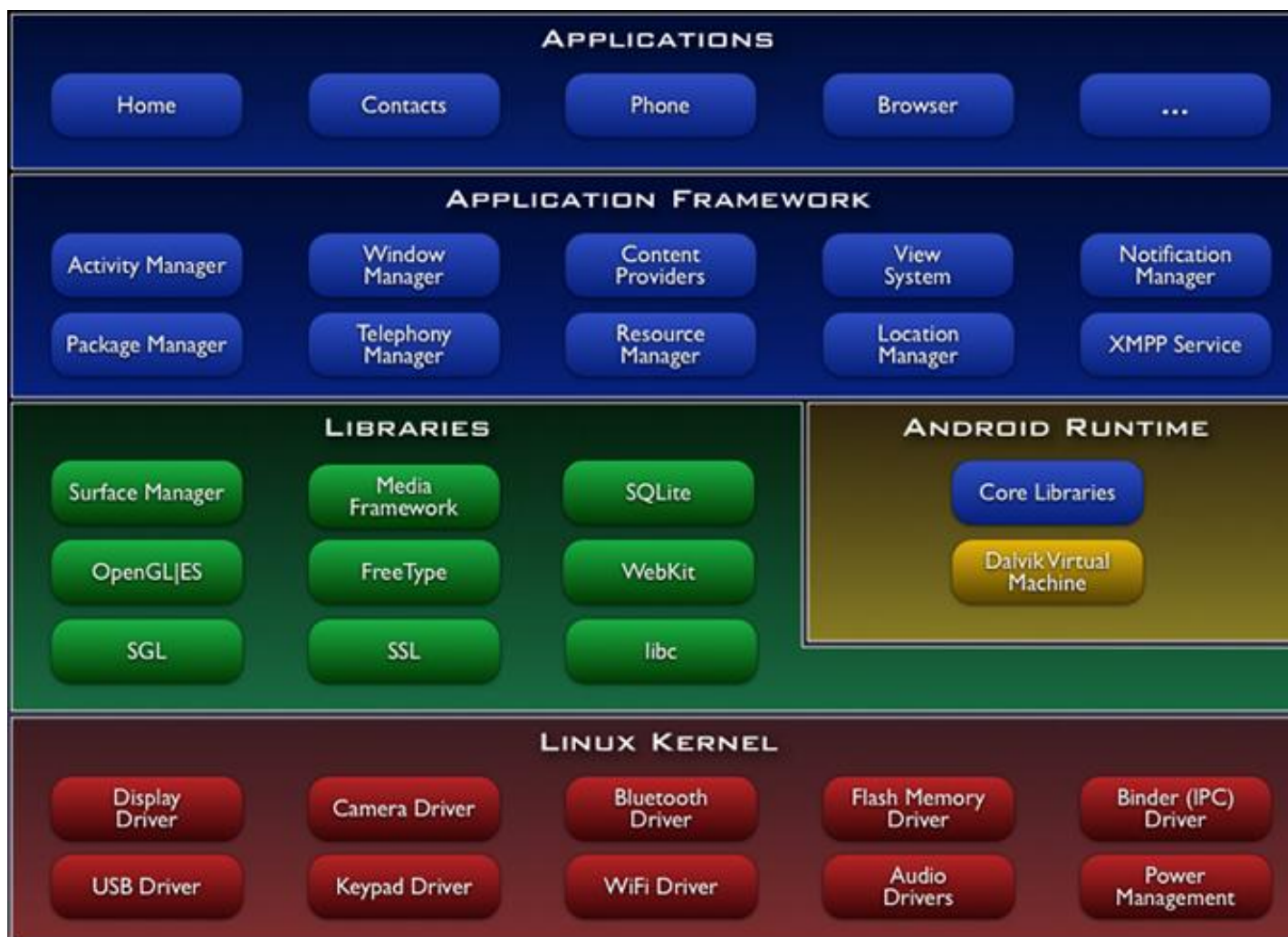


- } 华清远见嵌入式培训中心版权所有；
- } 未经华清远见明确许可，不能为任何目的以任何形式复制或传播此文档的任何部分；
- } 本文档包含的信息如有更改，恕不另行通知；
- } 保留所有权利。

内容提纲

- } Google Android 软件架构介绍
 - } Android 架构框图
 - } Android 系统源代码目录结构
 - } Android主要主要源码简介
- } 移植Android到新的硬件平台
 - } 移植Android到新硬件平台，需要的工作
 - } Android驱动编写
 - } Android 移植要点
 - } Android 移植流程
 - } Android 硬件系统要求
- NDK开发简介

Google Android 软件架构



Google Android 软件架构 (cont.)

- } Android系统架构和其操作系统一样，采用了分层的架构。从架构图看，Android系统架构分为四个层，从高层到低层分别为
 - } 应用程序层、
 - } 应用程序框架层、
 - } 系统运行库层
 - } linux核心层。



应用程序层

- } Android会同一系列核心应用程序包一起发布，该应用程序包包括email客户端，SMS短消息程序，日历，地图，浏览器，联系人管理程序等。所有的应用程序都是使用JAVA语言编写的。



应用程序框架

- } 开发人员可以完全访问核心应用程序所使用的API框架。
- } 隐藏在每个应用后面的是一系列的服务和系统, 其中包括:
 - } 丰富而又可扩展的视图 (Views), 可以用来构建应用程序, 它包括列表 (lists), 网格 (grids), 文本框 (text boxes), 按钮 (buttons), 甚至可嵌入的web浏览器。
 - } 内容提供者 (Content Providers) 使得应用程序可以访问另一个应用程序的数据 (如联系人数据库), 或者共享它们自己的数据
 - } 资源管理器 (Resource Manager) 提供 非代码资源的访问, 如本地字符串, 图形, 和布局文件 (layout files) 。
 - } 通知管理器 (Notification Manager) 使得应用程序可以在状态栏中显示自定义的提示信息。
 - } 活动管理器 (Activity Manager) 用来管理应用程序生命周期并提供常用的导航回退功能。有关更多的细节和怎样从头写一个应用程序, 请参考如何编写一个 Android 应用程序.



系统运行库

- } **Android系统架构** 包含一些C/C++库，这些库能被Android系统中不同的组件使用。它们通过 **Android 应用程序框架** 为开发者提供服务。以下是一些核心库：
 - } **系统 C 库**：一个从 BSD 继承来的标准 C 系统函数库（libc），它是专门为基于 embedded linux 的设备定制的。
 - } **媒体库**：基于 PacketVideo OpenCORE；该库支持多种常用的音频、视频格式回放和录制，同时支持静态图像文件。编码格式包括MPEG4, H.264, MP3, AAC, AMR, JPG, PNG。
 - } **Surface Manager**：对显示子系统的管理，并且为多个应用程序提供了2D和3D图层的无缝融合。
 - } **LibWebCore**：一个最新的web浏览器引擎用，支持Android浏览器和一个可嵌入的web视图。
 - } **SGL**：底层的2D图形引擎
 - } **3D libraries**：基于OpenGL ES 1.0 APIs实现；该库可以使用硬件 3D加速（如果可用）或者使用高度优化的3D软加速。
 - } **FreeType** -位图（bitmap）和矢量（vector）字体显示。
 - } **SQLite** - 一个对于所有应用程序可用，功能强劲的轻型关系型数据库引擎。

Android 运行库

- } Android系统架构包括了一个核心库，该核心库提供了JAVA编程语言核心库的大多数功能。每一个Android应用程序都在它自己的进程中运行，都拥有一个独立的Dalvik虚拟机实例。Dalvik被设计成一个设备可以同时高效地运行多个虚拟系统。Dalvik虚拟机执行（.dex）的Dalvik可执行文件，该格式文件针对小内存使用做了优化。
 - } 同时虚拟机是基于寄存器的，所有的类都经由JAVA编译器编译，然后通过SDK中的 "dx" 工具转化成.dex格式由虚拟机执行。Dalvik虚拟机依赖于linux内核的一些功能，比如线程机制和底层内存管理机制。
-



Linux 内核

- } Android 的核心系统服务依赖于 Linux 2.6 内核，如安全性，内存管理，进程管理，网络协议栈和驱动模型。Linux 内核也同时作为硬件和软件栈之间的抽象层。



Android 系统源代码目录结构

Project	Description
bionic	C runtime: libc, libm, libdl, dynamic linker Bionic含义为仿生，这里面是一些基础的库的源代码
bootloader/legacy	Bootloader reference code 启动引导相关代码
build	Build system build目录中的内容不是目标所用的代码，而是编译和配置所需要的脚本和工具
cts	Android兼容性测试套件标准
dalvik	Dalvik virtual machine JAVA虚拟机
development	High-level development and debugging tools 程序开发所需要的模板和工具
frameworks/base	Core Android app framework libraries 目标机器使用的一些库
frameworks/policies/base	Framework configuration policies 应用程序的框架层
hardware/libhardware	Hardware abstraction library 与硬件相关的库
hardware/ril	Radio interface layer
out	编译完成后的代码输出与此目录

Android 系统源代码目录结构(cont.)

kernel	Linux kernel Linux2.6的源代码
prebuilt	Binaries to support Linux and Mac OS builds x86和arm架构下预编译的一些资源
packages	Android的各种应用程序
sdk	sdk及模拟器
recovery	System recovery environment 与目标的恢复功能相关
system	Android的底层的一些库
vendor	厂商定制代码

移植Android到新硬件平台

移植Android到新硬件平台，需要的工作

GCC工具链

- 需要交叉编译工具链，如arm-eabi-4.2.1

Android操作系统内核

- 需要支持新硬件的Android Linux kernel

Hardware Abstraction layer

添加相关硬件抽象层code。

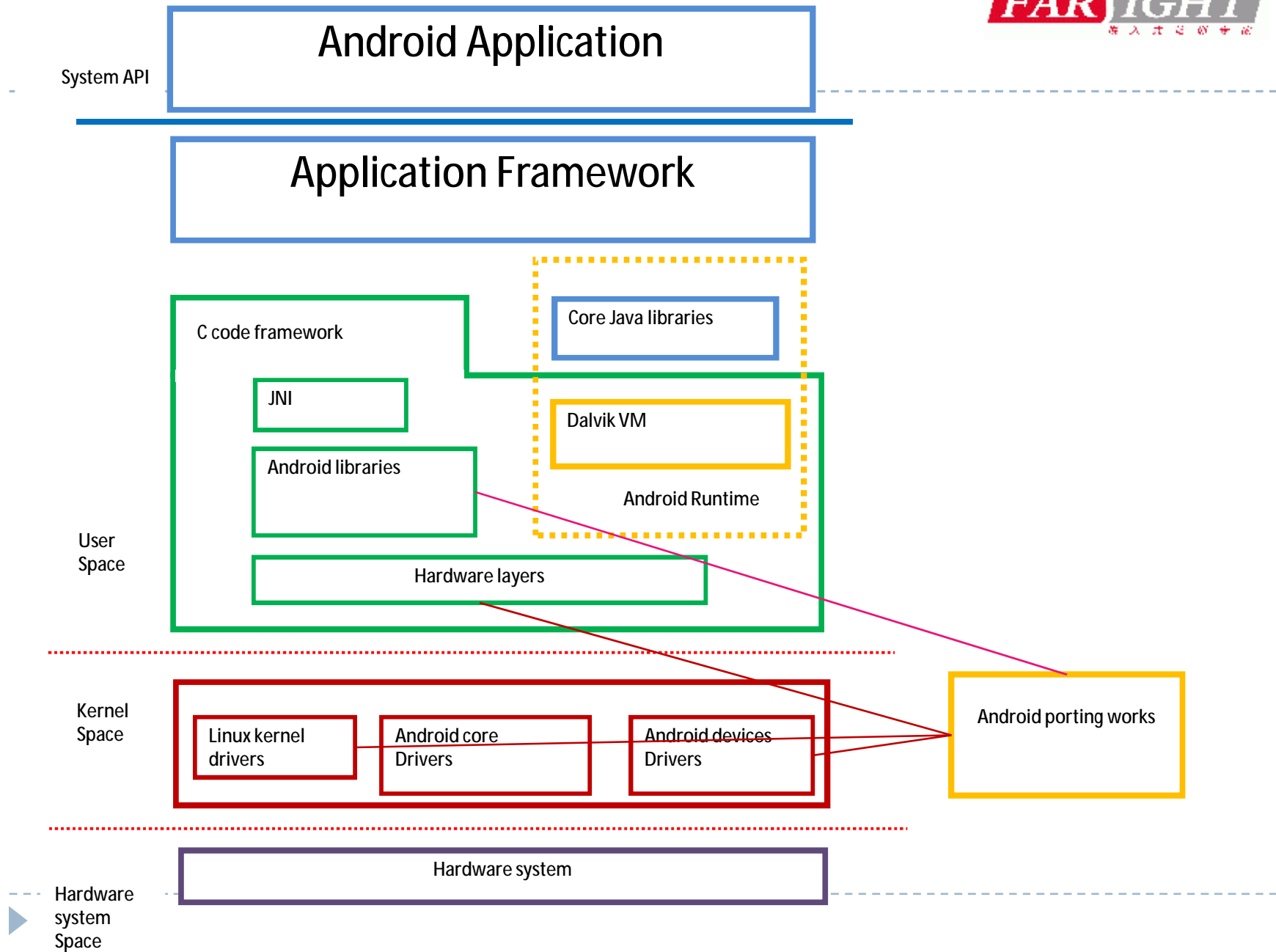
Dalvik VM / bionic

- 添加相关的移植和优化。

文件系统

- 添加system init相关。





移植要点

} GCC porting

} 工作重点是IC厂商和平台提供商。

} Dalvik VM / Bionic Porting

} 1.IC厂商和平台提供商

} 2.优化

} Linux Kernel Porting 方法:

} 1. 使用Android kernel

} 2.使用自己的kernel

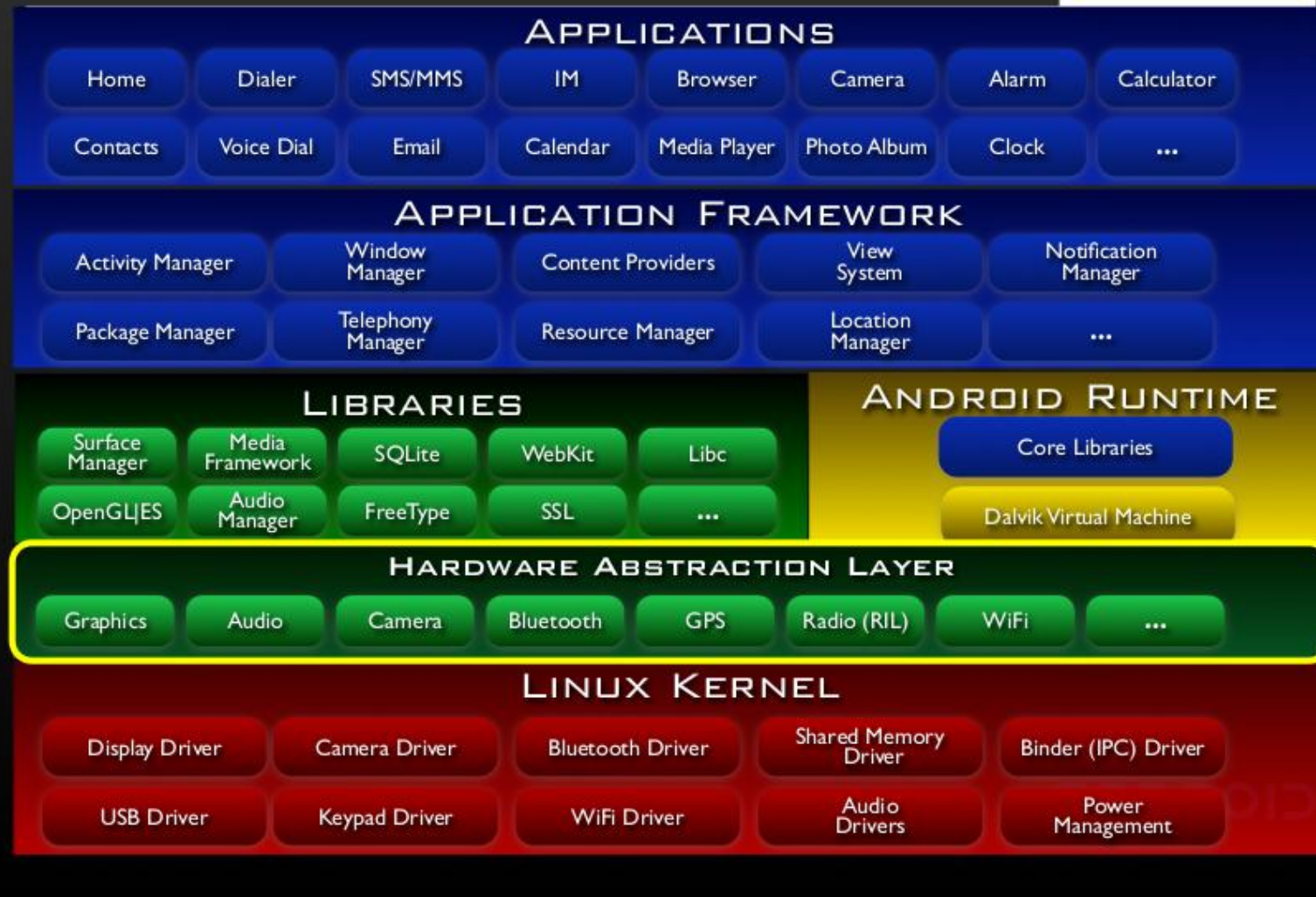
} Hardware Abstraction Layer Porting

} HAL 架构

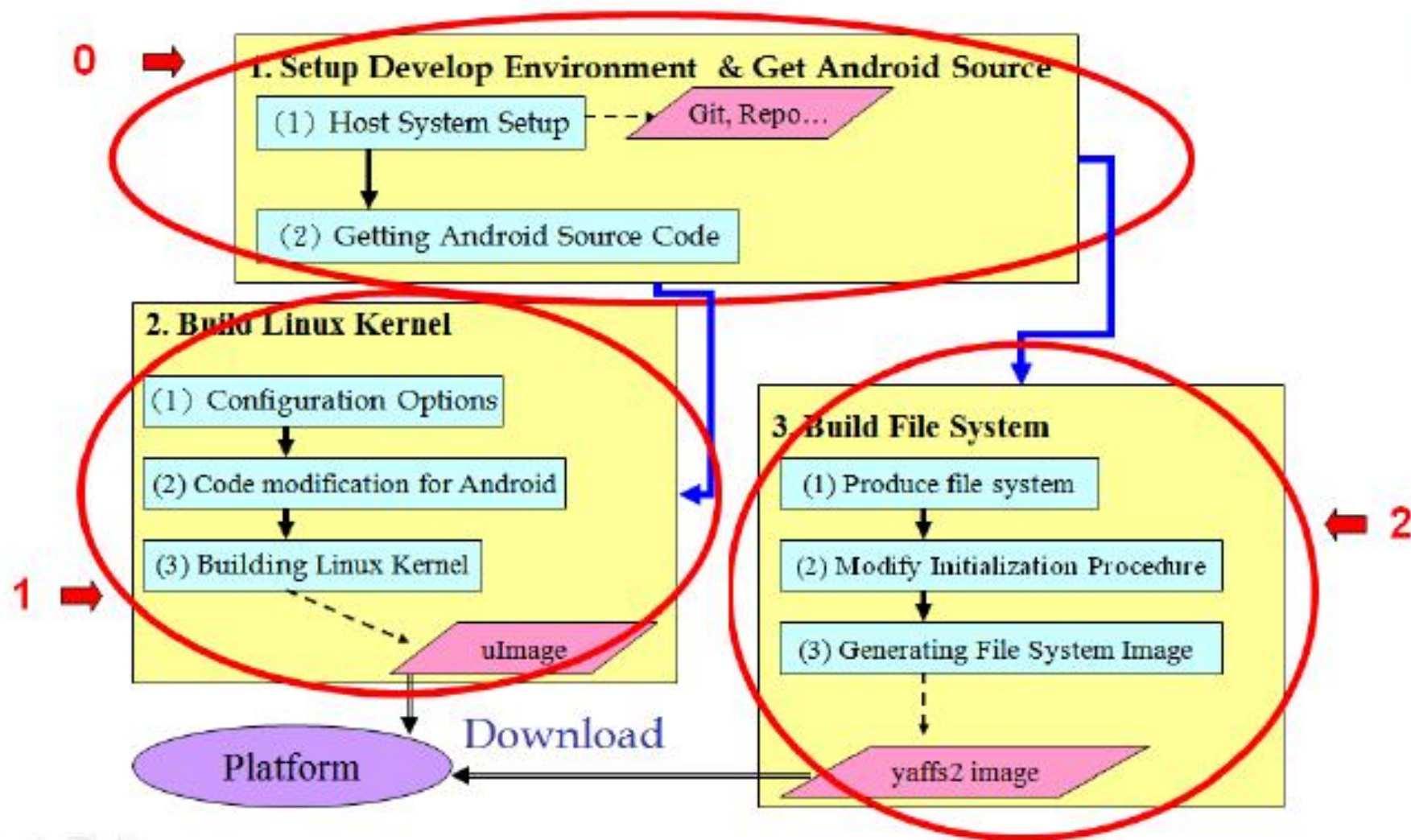




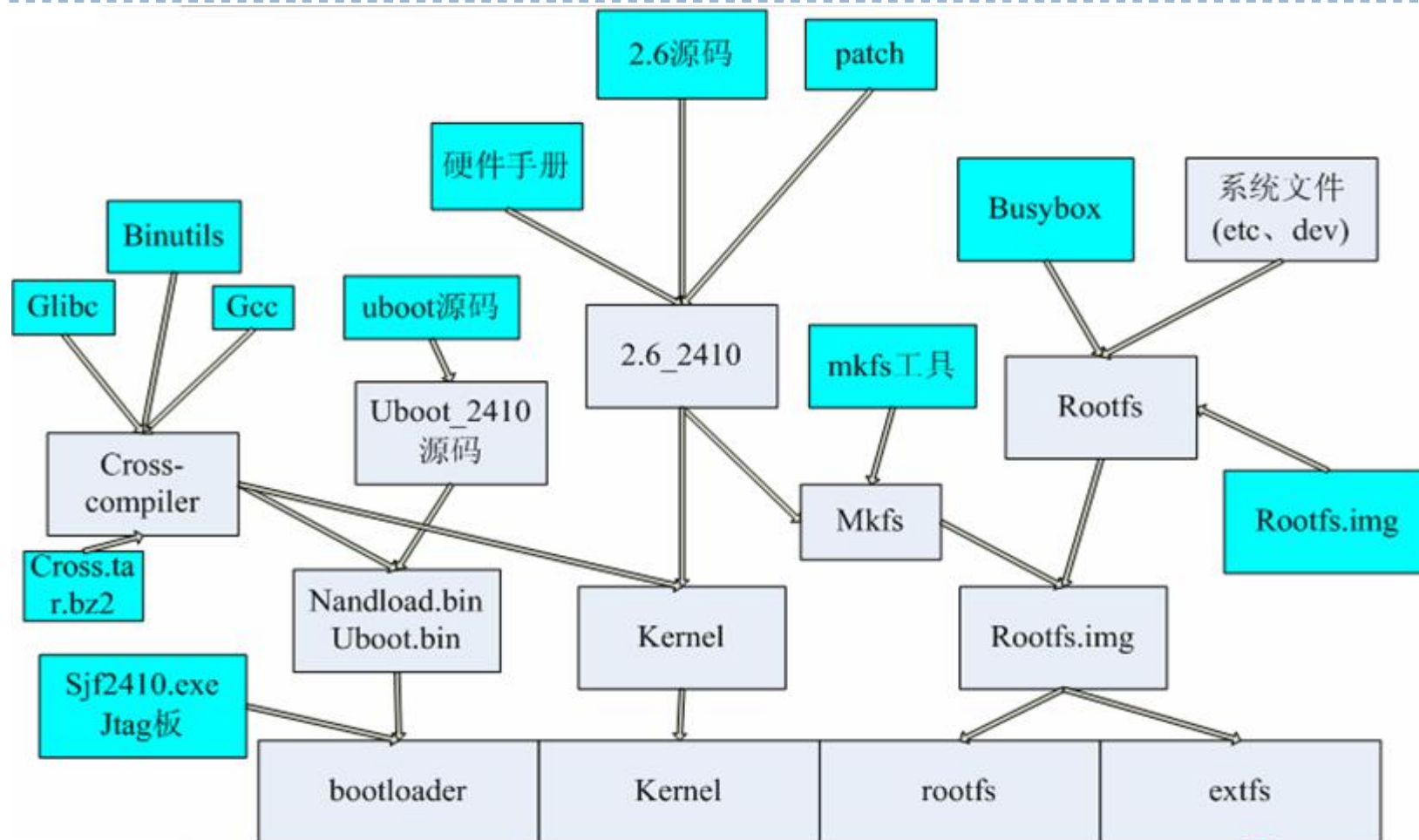
Hardware Abstraction Layer



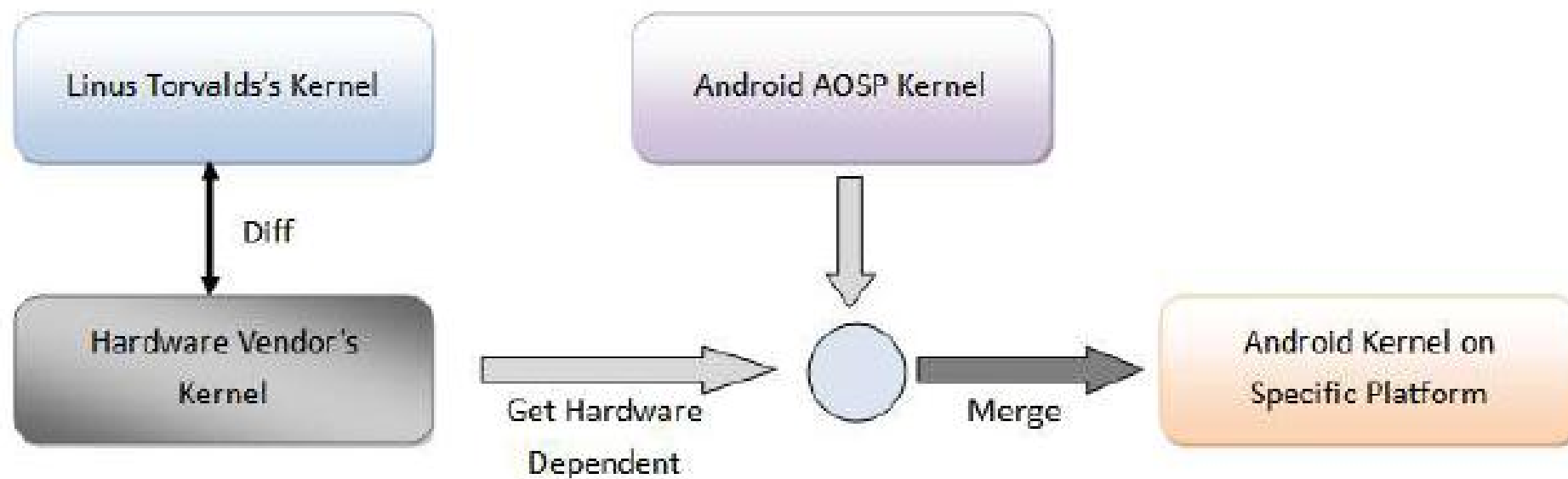
Android 移植流程



Linux系统的基本组成和开发流程图



Linux kernel 移植流程



Android 硬件系统要求



Feature	Minimum Requirement	Notes
Chipset	ARM-based	For the first release, Android is primarily targeted towards mobile handsets and portions of the platform, such as Dalvik VM graphics processing, currently assume an ARM architecture.
Memory	128 MB RAM; 256 MB Flash External	Android can boot and run in configurations with less memory, but it isn't recommended.
Storage	Mini or Micro SD	Not necessary for basic bring up, but recommended.
Primary Display	QVGA TFT LCD or larger, 16-bit color or better	The current Android interface targets a touch-based HVGA resolution display with a touch-interface no smaller than 2.8 inches in size. However, smaller displays will suffice for initial porting.
Navigation Keys	5-way navigation with 5 application keys, power, camera and volume controls	

Android 硬件系统要求 (cont.)



Camera	2MP CMOS	Not required for basic bring up.
USB	Standard mini-B USB interface	Android uses the USB interface for flashing the device system images and debugging a running device.
Bluetooth	1.2 or 2.0	Not required for initial bring up.

If available, your Android device can also benefit from the following optional device characteristics:

QWERTY keyboard

WiFi

GPS



Linux 内核的特性

- } 可以移植，支持的硬件平台广泛
 - } 有MMU和没有MMU的处理器均支持
 - } 32 bit 处理器 (arch/ subdirectories)
 - } alpha, arm, cris, frv, h8300, i386, m68k, m32r, m68knommu, mips, parisc, ppc, s390, sh, sparc, um, v850
 - } 64 处理器:
 - } ia64, mips64, ppc64, sh64, sparc64, x86_64
 - } 更多的细节看Documentation/<arch>/ （arch为处理器类型如 "arm"）
- } 高可扩展性
 - } 可剪裁、可扩展，可以运行在大型主机，也可以运行在个人计算机上
- } 高可靠性、稳定性
 - } 稳定性是linux鲜明特点，安装了linux系统的主机，连续运行一年不宕机是很平常的事情
- } 超强的网络功能
- } 真正的多任务，多用户系统
 - } 多个用户可以同时登录到系统同时工作
- } 耗费的资源相对较少
- } 安全性？
- } 模块化设计
 - } 模块可以动态加载，卸载，可以减少系统体积，同时可以用来解决冲突问题，模块调试
- } 遵循GPL开源许可协议的，开放系统
- } 编程更加简单，资源丰富

结 束

华清远见