

将AOSP移植到RISC-V平台：有多难？

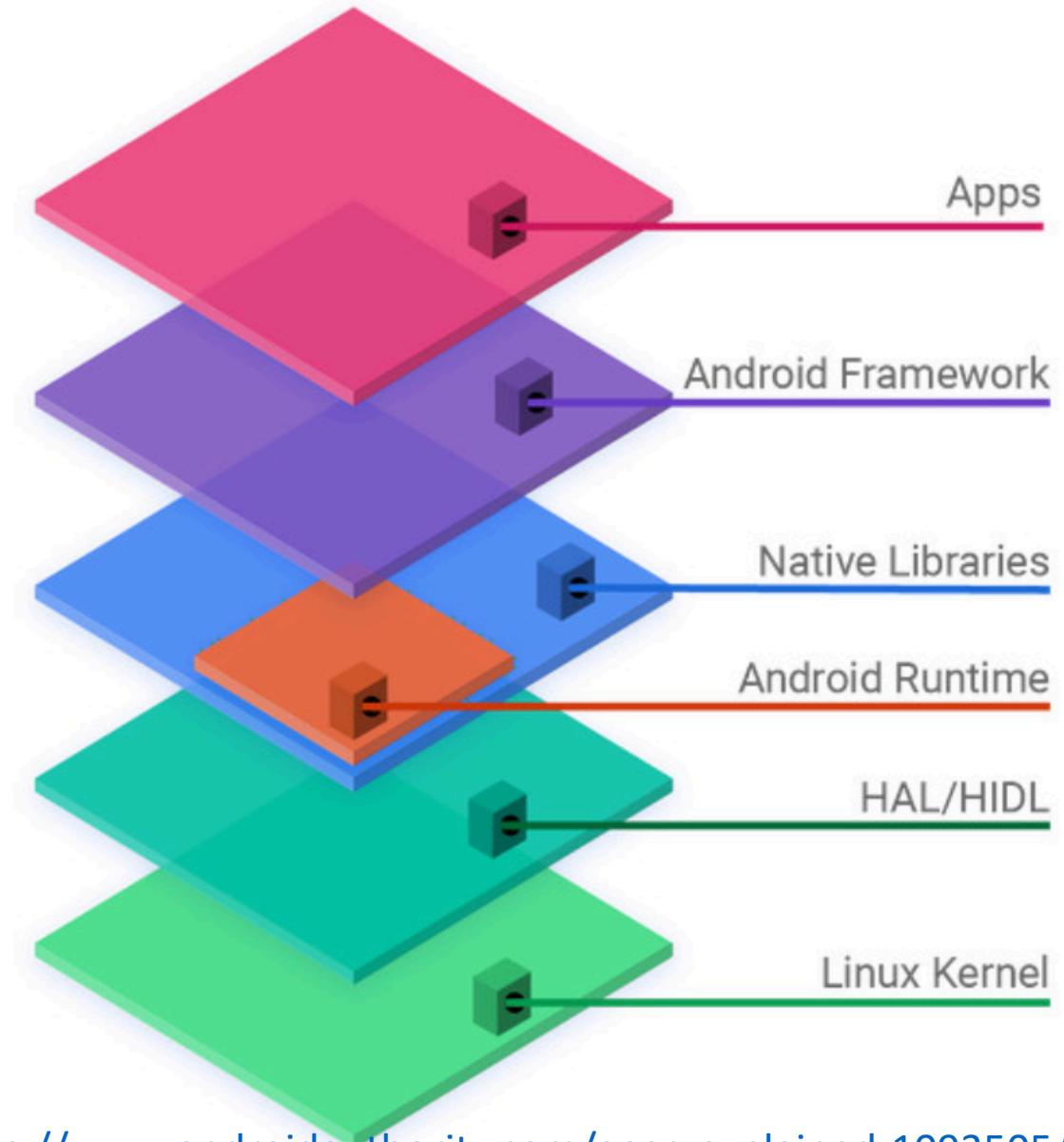
难 = 工程量 \times 技术难度 \times 偏门程度

吴伟 | @lazyparser | lazyparser@gmail.com

2020-06-03

AOSP: 热身知识介绍

- 我们都知道Android是什么
- 我们都知道Android重要性
- 架构信息网上非常多
- 文档做的也很好（尤其比国内的.....）
- Just Google it

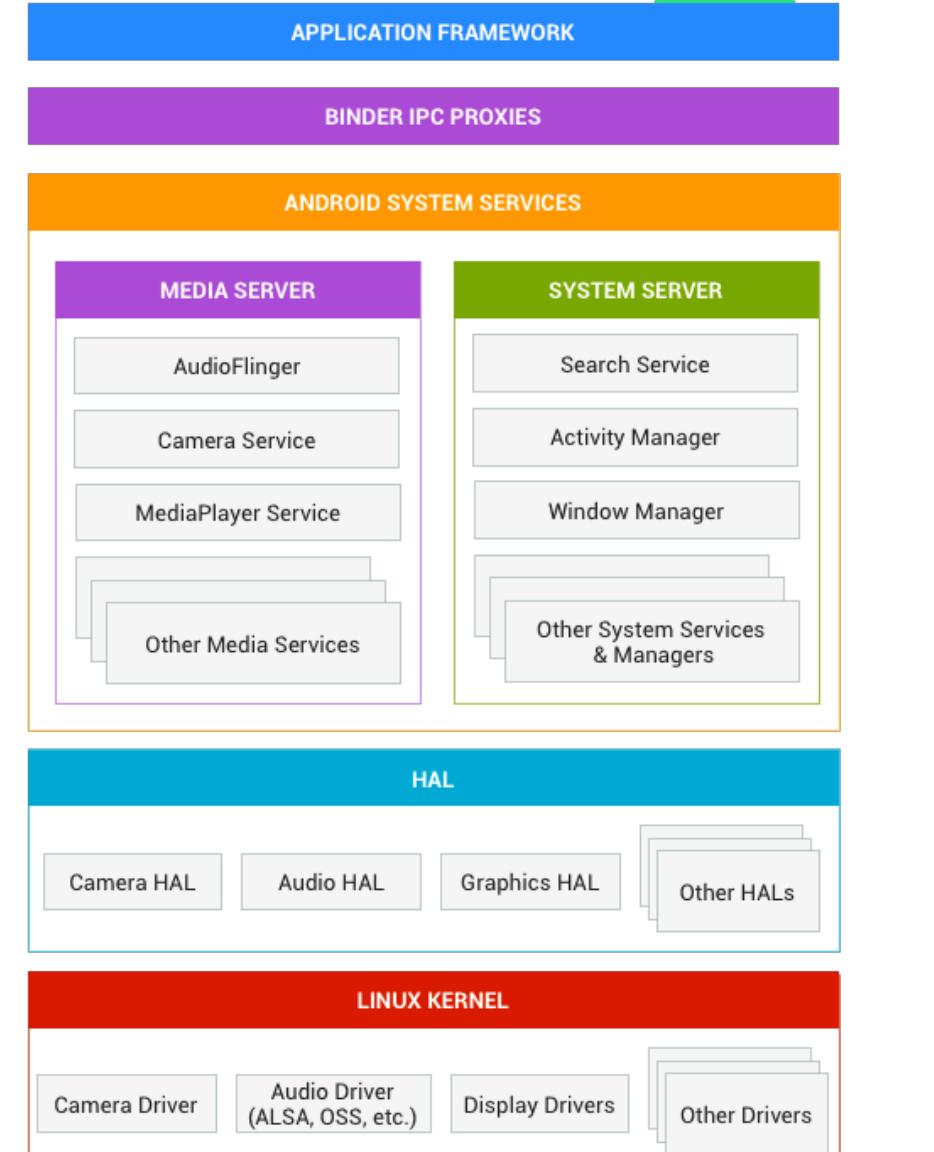


<https://www.androidauthority.com/aosp-explained-1093505/>



Android Architecture

- 越往上也不用考虑
- Java框架层面几乎不用考虑（真的？）
- C/C++系统库层面：交叉编译、部分需要移植
- Dalvik/ART、WebKit/Blink/V8
- 定制 Linux Kernel、一堆闭源二进制固件驱动



<https://source.android.com/devices/architecture>

Porting to RISC-V: Not Yet? Really? Why? Why?

- <https://groups.google.com/a/groups.riscv.org/forum/#!topic/sw-dev/tuANnW1uSsk>
 - 莫名其妙的变成LICENSE口水战，并没有什么价值。2019年5月
- <https://news.ycombinator.com/item?id=20525274>
 - It's mostly up to OpenJDK and V8 porting. some parts of the OpenJDK
 - Android does not use OpenJDK. It uses the ART runtime which will definitely need porting.
 - On the software side, Bionic (android's libc), Dalvik (runtime), LLVM, external dependencies (i.e. boringssl, lib{vpx,hecv,mpeg4,etc},) definitely need work.

那么、谁是Android的亲儿子？

- ARM32、ARM64：亲生儿子、唯一的皇位继承者
- X86：一直想上位
- MIPS：很努力了、真的很努力了 _(:_」∠)_
- Andes：相当有前途和潜力、可惜后来RISC-V出来了
- RISC-V：幺子 ^_(ツ)_/^- (微笑中透露出主角光环的从容)
- PowerPC：？？？认真的么

其他架构的兄弟在哪里？X86能刚arm了么？

- <https://www.android-x86.org/changelog.html>
 - 一晃十几年过去了
- <https://www.slideshare.net/RevistaSG/11-12-keynote-optimizacin-de-aplicaciones-android-para-x86>
 - ! 不是英文也不是中文
- https://coscup.org/2010/slides/15_3_1700_Android_x86_Project.pdf
 - 感谢宝岛的同行

X86 vs RISC-V：从大哥的失败未成功学到什么

- X86有着非常完善的工具链。而且是很多套
 - AOSP的x86的porting 都有很多团队在重复地做
- RISC-V LLVM 还在超着工业级稳定的方向上努力
- RISC-V GCC 要好一点
 - 而目前Android默认使用clang/llvm构建了

其他架构的兄弟们在哪里？MIPS？龙芯？

- <https://www.mips.com/develop/android/>
- 很努力了；只做到了断断续续的版本支持；
- MIPS的工具链等家底殷实
- 更重要的，这么多年大学教材御用，劳动力很多

其他架构的兄弟们在哪里？啥， PPC？

- <https://www.embedded.com/porting-android-to-the-powerpc-architecture/>
- **Step-1** : Kernel modifications for XPedite 6101 board and Android platform
- **Step-2** : Toolchain development for PowerPC platform
- **Step-3** : Android Open Source Project (AOSP) modifications for PowerPC platform
- 值得想要进行新架构的同行阅读
- https://elinux.org/images/0/07/Android_On_Power.pdf

其他架构的兄弟们在哪里？啥， PPC？

- <https://www.embedded.com/porting-android-to-the-powerpc-architecture/>

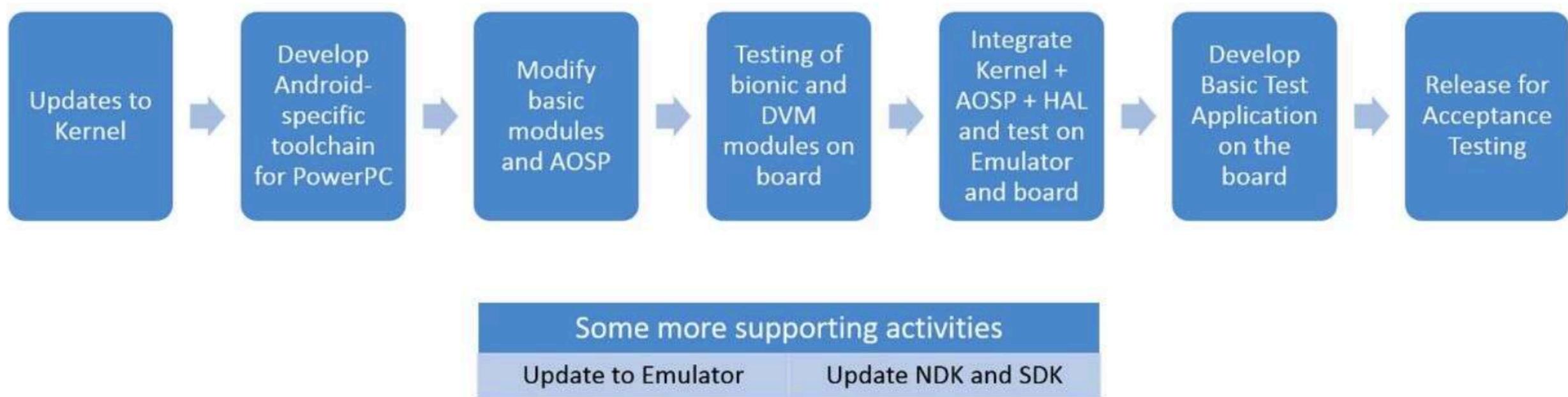


Figure 1: Step-by-step procedure depicting Android porting on PowerPC (Source: eInfochips)

最为慷慨和专业的分享，来自Andes的 Luse Cheng

- <https://www.slideshare.net/lusecheng/coscup2011-porting-android-to-brandnew-cpu-architecture>

• 在2011年的世界，
上述幻灯片的信息
至少价值20wRMB



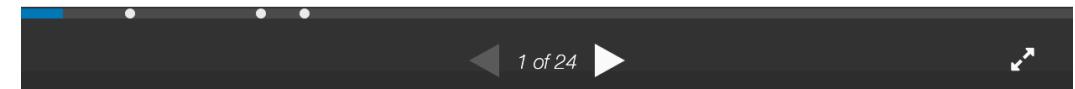
前浪大佬们的血泪（就不截图了，自己看吧）

<https://www.slideshare.net/linaroorg/porting-android-tonewplatforms>

4 people clipped this slide

Porting Android to new Platforms

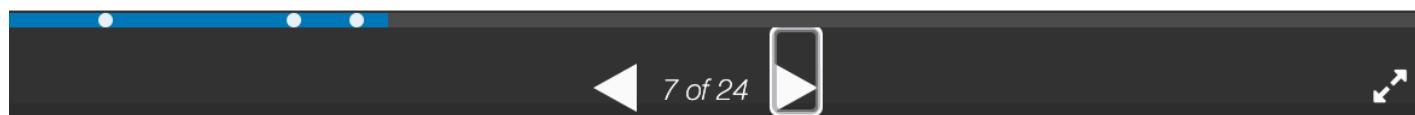
Amit Pundir
Android Engineer, Linaro



前浪大佬们的血泪

<https://www.slideshare.net/linaroorg/porting-android-tonewplatforms>

bionic	Android's C library.
bootable	Legacy Bootloader/Recovery code for reference
build	AOSP Build framework
cts	Android's Compatibility Test Suit
dalvik	Dalvik Virtual Machine
development	Development tools #tutorials #monkey
device	Vendor specific configuration files
external	External projects used in AOSP
frameworks	Android core application framework
hardware	Hardware specific libraries/modules #HAL
kernel	Linux Kernel
libcore	Apache Harmony. Fee Java implementation. Used by Dalvik.
ndk	Native Development Kit
packages	AOSP's stock applications
prebuilt	Prebuilt binaries such as Toolchain
sdk	Android's Software Development Kit
system	Android's core system libraries/binaries #init #toolbox #adb



2010年的乐观

Chip support

- armv5te and armv7a out-of-the-box
- MIPS, PPC, x86, SH ports have been done
- SoC support (kernel and drivers) from vendors including
 - Freescale i.MX, TI OMAP
- Custom board level support
 - In most cases you will have to create your own Android BSP
 - Requires merging Android kernel patches into your own kernel tree

Porting Android

5

- <https://elinux.org/images/7/71/03-android-inside.pdf>

有没有专门做 (Android|RV) Porting 的团队 ?

- <https://www.embien.com/android-porting-services>
- 中科创达 ?
- Andes (可能不接外包)
- CodeSourcey ?
- SiFive or StarFive ?

PLCT 能做到么？如何做？

- PLCT类似突击队或者特种小队，无法处理规模化的工作任务
- 第一步需要由OS发行版构建小队进行全系统构建，建立基线，建立基础设施
 - 目前1720+个仓库中，哪些可以用RV工具链构建通过，哪些失败；逐一分析
- 可能需要工具链小队先行处理RV工具链在构建中暴露的bugs和未实现feature
- 需要确认 Linux Kernel、BSP 层面的可用性，需要 Kernel/BSP 小队出来打
- 基础库部分、编译问题交给发行版小队，JIT问题可以交给PLCT/虚拟机小队
- Dalvik/ART 等部分，需要组建专门的Runtime开发团队进行死磕
- IDE依赖的工具链调试工具部分（SDK）需要有IDE小队和开发工具小队支持
- IDE中使用的模拟器需要模拟器小队开发或魔改

规模估计和时间估计

- Beyond Our Capabilities
 - 不过大体的时间估计和roadmap可以交流讨论
- 需要比PLCT更大规模的混合团队来一起做
- 需要有比 lazyparser 更为高阶的架构师/总工来进行任务分析和分解

<https://github.com/lazyparser/talks>

将AOSP移植到RISC-V平台：让我们开始吧

全世界 RISC-V 开发者、联合起来 (^o^)/

吴伟 | @lazyparser | lazyparser@gmail.com

2020-06-03